06242-6001-R001

FUNCTIONAL SPECIFICATIONS OF NRTPOD PROGRAM MODIFICATIONS

10 NOVEMBER 1966
Revised February 1967

Prepared for

Massachusetts Institute of Technology
Lincoln Laboratory

Under Contract No. CC 939

TRW SYSTEMS
AN OPERATING GROUP OF TRW INC.

06242-6001-R001

FUNCTIONAL SPECIFICATIONS OF NRTPOD PROGRAM MODIFICATIONS

10 NOVEMBER 1966
Revised February 1967

Prepared for

Massachusetts Institute of Technology
Lincoln Laboratory

Under Contract No. CC 939
PRIME CONTRACT AF 19(628)-5167

TRW SYSTEMS
AN OPERATING GROUP OF TRW INC.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

1. INTRODUCTION

This document describes the six modifications which have been added to the NRTPOD program in the form of extended capabilities. In addition, a separate stand-alone program which derives a preliminary estimate of an orbit and which is designed to be used in conjunction with NRTPOD is described in this document.

For purposes of distinction within this document the modified NRTPOD program is designated NRTPD2. However, it should be emphasized that this term is used to differentiate between different versions of the same program only; the subroutine descriptions still refer to the main program as NRTPOD.

This report is intended to be an analyst's guide to the NRTPOD modifications as well as an operational handbook with input/output instructions. An entire section of this report is devoted to an analytical and operational discussion of the six modifications which have been added to the NRTPOD program. The mathematical derivations of the various program options are not given completely in this section, although pertinent references (in which complete derivations are presented) are given in each respective section. The program descriptions given in this document are intended to be supplemental to the mathematical and operational descriptions of NRTPOD as described in Reference 1, Functional Specifications of Lincoln Laboratory Orbit Determination Programs. Since NRTPOD is a derivative of the ESPOD Program, considerable reference is made to the original ESPOD documentation, Reference 2.

In addition to the functional specifications of the NRTPOD modifications and the input/output instructions, a section of new and modified subroutines has been included. The standard description and accompanying flow diagrams of the modified subroutines have been abbreviated or eliminated in many instances in order to keep the volume of documentation to a manageable level. In these instances, reference to the main source of subroutine documentation is given.

PREPOD, a preliminary orbit determination program which derives an initial estimate of an orbit, is described in the last section of this report. Since this program is distinct from NRTPOD, the complete input/output and subroutine descriptions are given in their entirety in one section. The orbit determination method depends on the availability of two or more position fixes from an observing station over some free-flight arc. The preliminary conditions are derived by fitting an orthogonal polynomial to the components (topocentric range, azimuth, and elevation) of the position fixes.

The six modifications which have been added to NRTPOD are described briefly below.

Parameterization of Drag

This modification enhances the simulation of drag forces and, in particular, the capability of reducing data from reentering vehicles. The ballistic parameter $C_D A/M$ is represented by an altitude dependent, tabular function which is linearly interpolated in a given altitude range. This modification allows the analyst to simulate and regress on as many as 15 functional (altitude) values of the ballistic parameter.

Functional Standard Deviation

A trajectory functional standard deviation has been incorporated into NRTPOD to allow the assignment of data weights as a function of topocentric range, the radar cross section of the vehicle, and the particular sensor-vehicle geometry. The functional standard deviation is added to the nominal standard deviation and is usually negligible for nominal tracking distances.

Diagonalized Covariance Matrix Output

Whenever the normal matrix update option is exercised, diagonalized covariance matrix information is printed automatically. The printed output consists of the square roots of the eigenvalues and associated eigenvectors of the position and

velocity partitions of the orbit plane (UVW) covariance matrix. In addition, the three sequential rotation angles to align the UVW axes with the principal axes of the error ellipsoid are output. Finally, the determinants of specific covariance matrices are printed.

$\ddot{R}$ Observable

The NRTPOD program has been modified to accept and process $\ddot{R}$ data in addition to its conventional observables. This modification requires no special operational procedures, since the new observable is processed in the same manner as the observables which were in the program originally.

Steering Ephemeris

A radar steering option has been incorporated into the trajectory link of NRTPOD. By calling this option, the computed observables of every sensor in the sensor input list based on the current trajectory are printed at specified intervals for the length of the trajectory propagation. In addition to printing the steering ephemeris of each sensor, the current altitude, ballistic coefficient, and atmospheric density at the specified intervals are printed.

Linear Constraints

The capability of imposing linear constraints on the solution variables has been added to NRTPOD. This modification permits the analyst to require that any one of the solution variables to be a linear combination of any of the others in accordance with the requirements of the physical problem. An example of a physical constraint that should be accounted for in the tracking problem would be the precise knowledge of the relative locations of two observing radar stations. In this case, the linear constraints formulation would force the relative station geometry to remain fixed throughout the differential correction process.

The above modifications were coded in the FORTRAN IV programing language. Checkout and final integration of the modifications into the NRTPOD program were done on the IBM 7094 computer at TRW Systems facilities. A double precision version of NRTPD2 has been installed on the IBM 360/67 computer using the level H FORTRAN compiler. In fact, the program can be installed with only minor system interface modifications on any computer which accepts the ASA Standard FORTRAN IV language.

Accepted for the **Air Force**
Franklin C. Hudson
Chief, Lincoln Laboratory Office

2. GENERAL DESCRIPTION AND OPERATING INSTRUCTIONS

This section is concerned with a functional and operational description of the NRTPOD modifications. It is assumed that the reader is already familiar with the basic operation of the NRTPOD program as described in Reference 1.

2.1 PARAMETERIZATION OF DRAG

The NRTPD2 drag model is represented by an altitude dependent, linearly interpolated, tabular function. Up to fifteen altitude bands, ranging from sea level up to altitudes where the mean free path of atmospheric molecules is larger than the maximum spacecraft dimension, can be simulated in the trajectory model. The ballistic parameter, $C_D A/m = C(h)$, is linearly interpolated in a given altitude range, and the differential correction regresses on the functional (altitude) values.

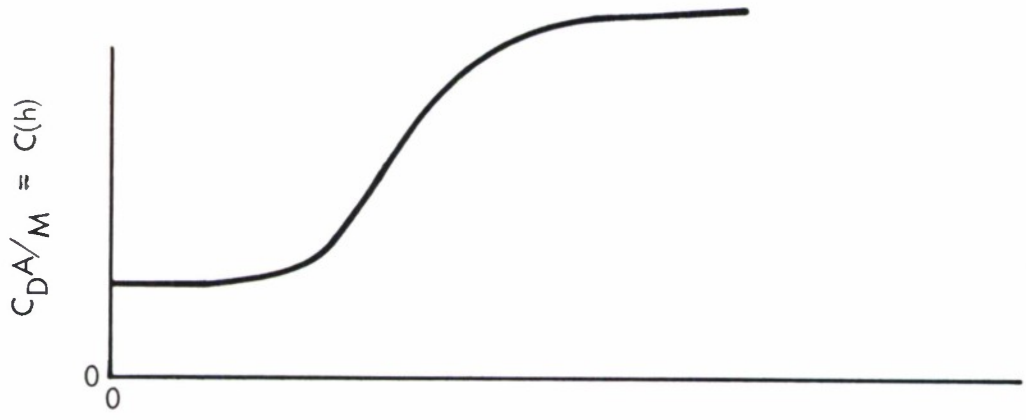


Figure 2-1. Representative Altitude Dependent Drag Model

The altitude is defined above the ellipsoid and used with the Lockheed-Jacchia atmosphere model. The required partial derivatives of the observations with respect to the ballistic solution parameters are obtained by simultaneously integrating the variational equations associated with the ballistic parameters (and hence, altitudes) bounding the vehicle at a particular time. Whenever an altitude range is crossed, the variational equations are reinitialized and the numerical integration is restarted. The detailed mathematical techniques associated with regression using the altitude dependent ballistic coefficients are treated in greater detail in Reference 3.

The linear constraints option discussed in Section 2.6 is a useful research tool to employ in conjunction with the variable drag model. It provides the analyst with flexibilities which the usual constraining techniques—that is, bounds—do not provide.

In an actual orbit determination using real data, the variable drag model should not be used initially. Since the drag variation has a relatively small effect on the trajectory, unless the vehicle is reentering the atmosphere, and considering that the larger solution vector enhances the computer running time appreciably, it is more efficient to regress for the position and velocity and possibly constant drag on the first determination with the real data. After bad observations have been rejected and a fairly good determination of the state vector has been established, the variable drag solution with the improved estimate of the state vector is attempted. This method is usually more successful than attempting to solve for the state vector and a variable drag model without a reasonably good estimate of the position and velocity of the vehicle.

The normal matrix generated during the differential correction can be updated to an arbitrary time by flagging the trajectory and update link fullowing a curve fit. The full normal matrix including the variable drag terms can be updated. The input requirements for the drag options and other operational considerations are treated in Section 3.3.2.

2.2 FUNCTIONAL STANDARD DEVIATION

A functional form of the standard deviation has been incorporated into NRTPD2. The standard deviation is a function of the trajectory relative to the sensor and the topocentric range. The functional form reduces essentially to the nominal standard deviation for the tracking of satellites in the normal operating range. However, as the topocentric range increases beyond the normal tracking distances, the range dependency results in a substantially higher standard deviation.

2.2.1 Mathematical Formulation

The functional form of the functional standard deviation is as follows

$$\sigma_f = \sqrt{A_i^2 + B_i R^4 / f(\theta)} \qquad (1)$$

where

A_i = nominal standard deviation

B_i = sensor dependent constant ($i = 1 \cdots 5$) denoting $A, E, R, \dot{R}, \ddot{R}$

R = topocentric range

$f(\theta)$ = radar cross section

θ = aspect angle

The quantity $f(\theta)$ is the radar cross section as a function of the angle away from the nose-on direction of an axially symmetric vehicle. The argument θ is the angle between the drag velocity vector and the topocentric range vector. This angle is trajectory dependent and its computation is derived in Reference 4. The function $f(\theta)$ is obtained by linear interpolation from the input table $\left[\theta_j, f(\theta)_j\right]$ $j = 1 \cdots n$, where $n \leq 7$; each table is sensor specific.

Graphically, the radar cross section function might appear as shown below, Figure 2-2.

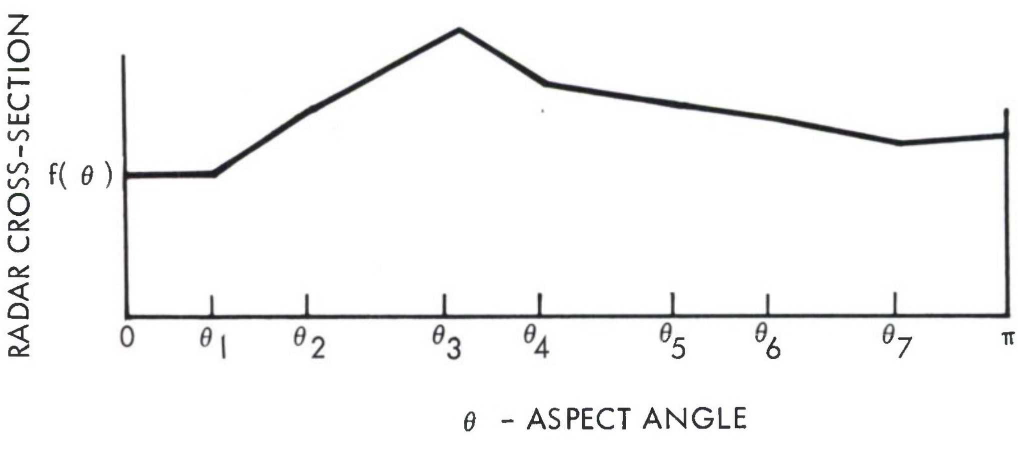


Figure 2-2. Representative Radar Cross-Section Profile as a Function of Aspect Angle

Given the argument θ (which is computed from the position and velocity of the spacecraft and the coordinates of the sensor) where $\theta_j < \theta \le \theta_{j+1}$ the corresponding value of $f(\theta)$ is obtained by linearly interpolating between $\left[\theta_j, f(\theta)_j\right]$ and $\left[\theta_{j+1}, f(\theta_{j+1})\right]$. The function is assumed constant between 0 and θ_1 and between θ_n and π. Mathematically

$$f(\theta) = \begin{cases} f(\theta_1), & \text{if } 0 \le \theta \le \theta_1 \\ f(\theta_n), & \text{if } 0 \le \theta \le \pi \end{cases}$$

If there is either one entry or none in the $[\theta, f(\theta)]$ table, the following conditions apply:

$$f(\theta) = \begin{cases} f(\theta_1), & \text{if } n = 1 \quad (0 \le \theta \le \pi) \\ 1, & \text{if } n = 0 \quad (0 \le \theta \le \pi) \end{cases}$$

2.2.2 Operational Specifications

The $[\theta, f(\theta)]$ table and the sensor constants B_i are input on sensor cards. See Section 3.4 for a detailed description of the sensor cards. The functional standard deviation option is flagged by the presence of a Type 4 sensor card (B_i constants) in the input deck. Even though the $[\theta, f(\theta)]$ table (Types 5 and 6 sensor cards) is not input, the functional standard deviation option is still operational under the assumption that $f(\theta) = 1$; see the previous section.

The sensor constants B_i are usually computed by the analyst given the tracking standard deviations as a function of topocentric range for a particular sensor. By rewriting the defining equation of the functional standard deviation as equation (2), a convenient method of calculating the sensor and observable dependent constants, B_i, can be illustrated.

$$B_i = (\sigma_f^2 - A_i^2)\, f(\theta)/R^4 \qquad (2)$$

Given the nominal standard deviation of a sensor, the radar constant can be evaluated if the functional standard deviation is specified for a satellite of known radar cross section and range. The radar constant is in mixed units as the individual terms in equation (2) are in set units. The standard deviations (both functional and nominal) are specified in card input units, as shown in Table 2-1 below.

Table 2-1. Units of Standard Deviations of Observables Used to Evaluate Sensor Constants

Observable	Units
Azimuth	Degrees
Elevation	Degrees
Range	Kilometers
Range-rate	Kilometers/second
Range acceleration	Meters/second/second

The radar cross section $f(\theta)$, is in $(\text{meters})^2$, and the specific range for which the constant is being evaluated is in earth-radii. For example, if the nominal standard deviation in range of a sensor is 40 meters, and the functional standard deviation is 50 meters for a satellite of 2 m^2 in radar cross section at 4,000 km range, the computation of B_R would be carried out as shown below:

(1 earth radius = 6378 km)

$$B_R = (\sigma_f^2 - A_i^2)\, f(\theta)/R^4$$

$$= \left[(0.05)^2 - (0.04)^2\right] (2.0)/(4000.0/6378.0)^4$$

$$B_R = 11.6 \times 10^{-3}$$

The A_i constant in the defining equation of the functional standard deviation is equal to the nominal standard deviation, which is the standard deviation the program uses if the functional form is not called. If the standard deviation is entered on both the observation card (σ_o) and the sensor card (σ_s), the observation card value is used.

Mathematically,

$$A_i = \begin{cases} \sigma_s, & \text{if } \sigma_o = 0 \\ \sigma_o, & \text{if } \sigma_o > 0 \end{cases}$$

Since the aspect angle, and hence the functional standard deviation, is trajectory dependent, the weight applied to a particular residual varies

from iteration to iteration. The standard deviations of each observation are printed as a companion page to the residuals print. See the sample output in Section 4.6. The functional standard deviations print is optional and is called by setting Column 43 of the JDC equal to one.

2.3 DIAGONALIZED COVARIANCE MATRIX OUTPUT

Diagonalized covariance matrices and associated quantities are output at each update (DELTT) time whenever a trajectory and matrix update are performed. No flags are required to obtain this output as it is automatically computed during the process of a matrix update.

The eigenvalues and associated eigenvectors of the upper 3 x 3 (position) and lower 3 x 3 (velocity) partitions of the UVW covariance matrix are computed at each update time. The UVW system is a vehicle-centered coordinate system; see Section 6.4 of Reference 1. The UVW covariance matrix is not an output quantity of NRTPD2, although it is internally computed from the Cartesian covariance matrix. The square roots of the eigenvalues (and the associated eigenvectors) are output; see Section 4.11 for a sample printout.

The orientation of the position and velocity error ellipsoids with respect to the U, V, W axes is such that the principal axes are identified by the nearest axes of the U, V, W set. From Figure 2-3 and on the assumption that the spin axis of the vehicle is coincident with the V axis (downrange direction), the sense of positive rotations can be derived. In this configuration, a rotation about the U axis results in a yaw of the vehicle; a positive rotation about U is defined as V x W, hence yaw positive is turning to the left. A rotation about the V axis is a roll maneuver; a positive rotation is W x U or clockwise. A rotation about the W axis results in a pitch of the vehicle (V) axis; positive pitch is defined as U x V, or down. The ordered rotations to align the UVW coordinates with the principal axes of the error ellipsoid for the NRTPD2 program will be sequential rotations about (1) the U axis, (2) the V axis, and (3) the W axis; i.e., yaw-roll-pitch. The ordered rotations and associated definitions are summarized in Table 2-2.

Table 2-2. Summary of UVW Coordinate System Rotation Conventions

	Spin Axis	Positive Rotation Definition	Alignment Nomenclature	Positive Direction	Angle
1	U	V x W	yaw	left	o_1
2	V	W x U	roll	clockwise	o_2
3	W	U x V	pitch	down	o_3

The ordered yaw-roll-pitch rotations of the U, V, W coordinate axes which will align them with the error ellipsoid are illustrated in Figure 2-3.

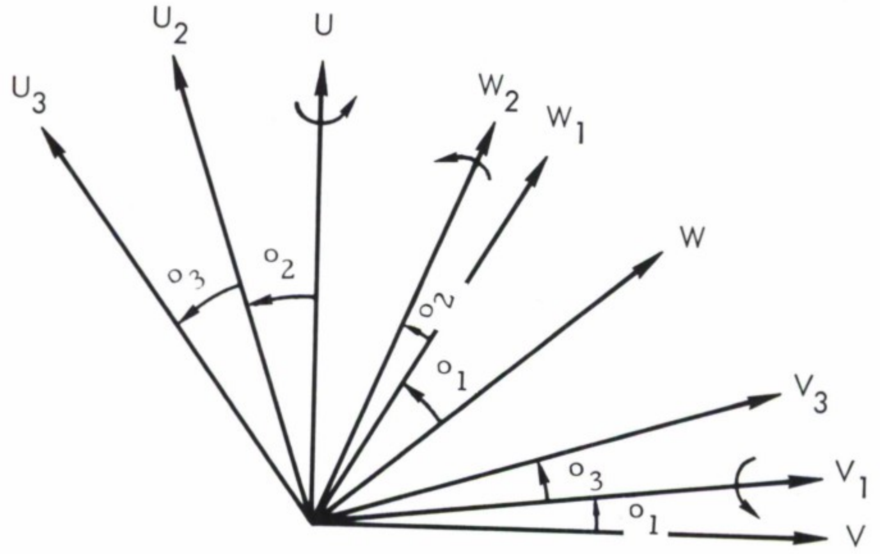


Figure 2-3. Sequential Rotations of the UVW Coordinate System to Align it with the Principal Axes.

In this illustration, according to the definitions in Table 2-2, yaw (about U) and roll (about V) are positive rotations and pitch (about W) is a negative rotation. For both the position and velocity partitions of the UVW

covariance matrix the ordered yaw-roll-pitch rotations of the U, V, W axes which will align them with the principal axes of the error ellipsoid are output.

In addition to the diagonalized covariance matrix output, the determinants of the following (sub) matrices are computed and output:

1) Square root of the determinant of the upper (position) 3 x 3 partition of the Cartesian covariance matrix (km^3)

2) Square root of the determinant of the lower (velocity) 3 x 3 partition of the Cartesian covariance matrix (km^3/sec^3)

3) Square root of the determinant of the 6 x 6 Cartesian covariance matrix (km^6/sec^3)

The Cartesian covariance matrix itself is not output; it is computed internally from the polar spherical (ADBARV) covariance matrix. The determinants are computed by a matrix decomposition method. The LEGS2 subroutine uses such a method in the process of solving the normal equations; hence, the subroutine was used to evaluate the determinants. See Reference 5 for a mathematical discussion of the evaluation of determinants as accomplished in this program. The output units of the square roots of the determinants are indicated above. See Section 4.11 for a sample output of the quantities described in this section.

2.4 $\ddot{R}$ DATA

Range acceleration has been added to the list of observables that are acceptable to NRTPD2. There are no specific instructions as to its use other than the input and output descriptions. (See Sections 3 and 4.) The only exception involves the solution of observation time bias. Since the partial derivative of range acceleration with respect to time is not included in the $\ddot{R}$ partials, it is not possible to regress for time bias when using $\ddot{R}$ data only.

The equations required to implement this modification may be found in the equation section of the pertinent subroutines. The modified subroutines are: DRDP, PRELIM, PUPB, RADR (Section 5). The mathematical formulation of the required $\ddot{R}$ partial derivatives and associated computations are developed in Reference 6.

2.5 STEERING EPHEMERIS

The radar steering package is a trajectory or post-differential correction option of NRTPD2. When the option is called, the following trajectory parameters are printed at each specified (DELTT) time:

1) h = Height (km) of the vehicle above the ellipsoid, as used for entry in the Lockheed-Jacchia atmosphere model.

2) $C(h) = C_D A/m$ (m^2/kg), the ballistic coefficient.

3) $\rho(h)$ = Atmospheric density (kg/m^3).

In addition, the following radar (topocentric) parameters are printed for each sensor in the Master Sensor Table:

4) A = Azimuth of the vehicle (degrees)

5) E = Elevation of the vehicle (degrees)

6) R = Range to the vehicle (km)

7) $\dot{R}$ = Range rate of the vehicle (km/min) or (km/sec)

8) $\ddot{R}$ = Range acceleration of the vehicle (km/min^2) or (m/sec^2)

The three trajectory only dependent parameters h, C(h), and $\rho(h)$ are computed in the atmosphere related subroutines, that is, DRAG, JACHIA, and ATM59. The sensor dependent parameters are computed in subroutine STEER. The equations for the radar steering ephemeris are developed in Reference 7.

The radar steering option is called by setting Column 54 of the JDC card. The steering information for a particular time is printed following the trajectory block. If it is desired, the trajectory print can be suppressed. The tabulation below summarizes the steering options.

JDC Column	Content	Description
54	0	No steering
54	1	Trajectory print and steering
54	2	Steering only

When a differential correction precedes a trajectory propagation and steering run, the radar ephemeris is based on the converged or best estimate of the trajectory at the termination of the differential correction. If the ballistic coefficient is included in the solution vector, it is transferred to the trajectory link also for trajectory and steering computations. If drag is not included in the solution vector or if there is no differential correction, initial estimates of the ballistic coefficient may be input on preliminary data cards. See Section 3.3.2 for a description of the preliminary data input cards which specify the drag model.

The steering ephemeris is constrained to print only when the vehicle is above a nominal horizon of -5°. Since there is no search for rise and set times, the steering output begins (ends) at the first (last) point above the horizon. The nominal horizon (-5°) may be changed with a preliminary input card— ECRIT (Section 3.3.3). The output units of some of the steering ephemeris parameters can be varied (Section 3.3.3).

2.6 LINEAR CONSTRAINTS

The linear constraints option provides the analyst with additional control over the solution vector. This method provides the capability of specifying the correction of a particular variable in terms of a linear combination of another variable, whereas the bounds technique only permits the analyst to specify the maximum correction to a solution variable on a given iteration.

2.6.1 Applications

Constraints among the solution variables are often a part of the physical problem. An example of a physical constraint with application to the tracking problem would be the precise knowledge of the relative locations of two radar stations. If the actual locations are among the solution variables in a differential correction, it is important to constrain the corrections so that the relative locations are preserved. If a nonlinear constraint is required, it is still possible to apply it to the solution variables, although the formulation will only be valid for one iteration. Keeping the orbital period constant is an example of a nonlinear constraint. In actual practice, linear constraints are widely used in the solution of sensor biases, especially if an orbit determination involves a particular tracking

net with similar tracking equipment. When certain parameters of a physical system which are usually uncorrelated are related, it is advantageous to account for it in the solution. For example, if pass-by-pass bias solutions are desired for a single radar station, a different station identification can be assigned to each pass and then, by linear constraints, require that the station location biases for each identification be equal.

2.6.2 Constraint Matrix

The constrained solution is implemented by introducing linear constraints of the form

$$x = By \tag{3}$$

where $x_{n \times 1}$ is the original solution variable, B is the constraint matrix, and $y_{m \times 1}$ is the reduced set of solution variables. Therefore, the constraint matrix B is a rectangular matrix ($n \times m$), $m \leq n$ whose elements describe the linear relation between the solution variables. The mathematical formulation for the implementation of linear constraints into NRTPD2 is developed in Reference 8.

2.6.3 Examples

As a first example, suppose a differential correction was being made for the initial conditions of the spacecraft, the station locations of stations AA and BB of known angular separation, and the angle biases of station BB. By constraining the changes to the station locations to be the same, the known angular separation will be preserved. The constraint matrix B would look like Figure 2-4., where the rows are the original solution variables and the columns, the constrained variables. The preliminary input which specifies this constrained solution consists of the two following cards: (See Section 3.3.4 for linear constraints input.)

BIJ = 101, 202, 303, 404, 505, 606, 707, 808, 909, 1010, 1107, 1208

XIJ = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

As was mentioned previously, linear constraints can be used to keep the energy of an orbit constant during a differential correction. A second example follows to familiarize the reader with the process of setting up a linear constraints case. From the energy equation

$$v^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right) \tag{4}$$

		INDEPENDENT SOLUTION VARIABLES									
		α	δ	β	A	r	V	AA BB o	AA BB λ	BB A_b	BB E_b
ALL SOLUTION VARIABLES	α	1									
	δ		1								
	β			1							
	A				1						
	r					1					
	V						1				
	AA o							1			
	AA λ								1		
	BB A_b									1	
	BB E_b										1
	BB o							1			
	BB λ								1		

Figure 2-4. Constraint Matrix

it is easily derived that, from the condition that the energy remain constant, the permissible change in velocity in terms of position and velocity is given by equation (5)

$$\Delta v = -\Delta r \left(\frac{\mu}{r^2 v}\right) \tag{5}$$

where μ is the gravitational constant.

Given

$$\mu = 5.5303934 \times 10^{-3} \text{ e.r.}^3/\text{min}^2$$

$$r = 9567.2475 \text{ km} = 1.50 \text{ earth radii (e.r.)}$$

$$v = 8.50422 \text{ km/sec} = 0.08 \text{ e.r./min}$$

Substituting into equation (5), one finds

$$\begin{aligned} \Delta v &= -\Delta r \frac{\mu}{(1.5)^2 (0.08)} \\ \Delta v &= -\Delta r\ (0.0307) \end{aligned} \tag{5a}$$

Hence, for this second example, the preliminary data input which controls the linear constraints option looks like the two input cards illustrated below

BIJ = 101, 202, 303, 404, 505, 605

XIJ = 1, 1, 1, 1, 1, -0.0307,

It should be noted that the energy constraint (that it should remain constant) is valid for the first iteration only since the value of the constant (μ/r^2v) changes with succeeding iterations. However, this is only true when the proportionality constant is a function of the solution variables, such as this constant energy constraint. Also, the proportionality constant is computed and input in internal (program) units; that is, earth radii and minutes.

2.6.4 Output

The printed output of a linear constraints case is nearly identical to the standard output. The normal matrix is accumulated in the constrained system and hence is output similarly; that is, the matrix is an m x m, where m is the number of solution variables in the constrained system. The variance-covariance matrix and the correlation matrix are output in terms of the original solution variables; that is, the dimension is n x n.

3. PROGRAM INPUT

New inputs have been provided for control of the options that have been added to NRTPOD. The program options and the respective input requirements are listed below.

Program Option	Input Requirements	Section
Parameterization of Drag	Preliminary data cards	3.3.2
Functional Standard Deviation	JDC card Sensor cards	3.2 3.4
Diagonalized Matrix Output	None	
$\ddot{R}$ Observable	Observation cards Sensor cards	3.5 3.4
Steering Ephemeris	JDC card Preliminary data cards Sensor cards	3.2 3.3.3 3.4
Linear Constraints	Preliminary data cards	3.3.4
Mean Elements Input	Preliminary data cards (Six-card element set)	3.3.1

3.1 DECK SET-UP

The input deck sequence of NRTPD2 is identical to the input sequence of NRTPOD, as described in Section 2.2.1 of Reference 1.

3.2 JDC—JOB DESCRIPTION CARD

The JDC card is the control card for the flow of information through NRTPD2. This card is always the first card of an input data deck. It selects certain program options and defines the program sections to be used. A short arbitrary remark is permitted on the card.

Column	Content	Description
1-3	JDC	Identifies JDC card
4-7		Vehicle number
8-17		Vehicle name
18-29		User's header
30	Not used at present	

Column	Content	Description
31	0 or blank	Sensor and observation data not to be processed
	1	Sensor and observation data to be processed
	2	Sensors only
32	0 or blank	Do not print sensor data
	1	Print sensor data
33	0 or blank	Do not print observations
	1	Print observations
34	0	Do not print functional standard deviation input
	1	Print functional standand deviation input
35	0	Observations not presorted, fewer than 345 cards.
	1	Observations presorted, no maximum
36-40	Not used at present	
41	0 or blank	Curve fit not desired
	1	Curve fit desired
42	0 or blank	_A priori_ S matrix _not_ input on this run
	1	_A priori_ S matrix is input on this run
43	0 or blank	Do not print functional standard deviations
	1	Print functional standard deviations
44-50	Not used at present	
51	0 or blank	Trajectory print not desired on this run
	1	Trajectory print is desired on this run
52	0 or blank	_A priori_ UPMAT (covariance) matrix not input
	1	_A priori_ UPMAT matrix is input on this run
53	Not used at present	

Column	Content	Description
54	0 or blank	Steering ephemeris not desired
	1	Print steering ephemeris and trajectory
	2	Print steering ephemeris only
55	0 or blank	Covariance matrix update not desired
	1	Covariance matrix update desired

3.3 PRELIMINARY DATA INPUT

3.3.1 Mean Elements Input

The NRTPD2 Program accepts Kozai mean elements for the specification of position and velocity of a satellite at a given time. The elements are input Type 4. The elements need not be referenced to the epoch of the run in question since the capability to update the input mean elements to an arbitrary epoch has been provided.

Mean elements in the revised SPADATS/SPACETRACK format are input. This is a six-card element set, which is described in detail in Section 1.2.4 of Reference 1. When mean elements are input (TYPE = 4), an ITIME card is not required as it is for the other types. The TNULL card is used to specify the epoch in conjunction with the mean elements cards.

The TNULL card specifies the time to which the mean elements are to be updated relative to the epoch given on the six-card element set. The updated epoch is the epoch of a given run, such as the time associated with the initial estimate of position and velocity of a satellite in a differential correction.

The epoch is given in Julian days on Card 2 of the six-card set. TNULL is a three entry array as shown below:

TNULL = DAYS., HOURS., MINUTES.,

or

TNULL(2) = HOURS., MINUTES.,

or

TNULL(3) = MINUTES.,

3.3.2 Parameterization of Drag

Four input variables are required to exercise the variable drag options of NRTPD2.

Variable Name	Description
ALTS	The ALTS card specifies the altitudes (in kilometers) of the ballistic coefficients as given on the CLAMDA card. The trajectory of the vehicle must be enveloped by the top and bottom drag layers; if the vehicle goes outside these regions, the program will exit with an error message.
CLAMDA	This card specifies the ballistic coefficients, C_DA/m (m^2/kg), in a one-to-one correspondence with the altitudes as specified on the ALTS card.
CATLM	The CATLM array indicates to the program the ballistic coefficients which are to be solved for. The first entry relates to the uppermost layer in the atmosphere. As in the CAT1 and CAT2 cards, a "1" indicates the variable is to be solved for and a "0" indicates the variable is <u>not</u> to be solved for. The order of the solution flags corresponds to the ordered entries of the ALTS and CLAMDA cards. See the following example.
CHEPS	This card is a single entry that specifies the altitude cut-off criterion. In the process of integrating a trajectory, the program iterates for the position and velocity of the vehicle as it crosses a defined layer. When a layer has been crossed, an iterative procedure is initiated; the nominal criterion for convergence is 10^{-6} earth-radii.

Example:

Given the following atmospheric drag model:

Altitude (km)	C_DA/m (m^2/kg)
120	.008
100	.006
90	.003
85	.001

To solve for the drag at the two lower altitudes, the input cards that specify the trajectory model and the solution vector would look as follows:

ALTS = 120., 100., 90., 85.,

CLAMDA = .008, .006, .003, .001,

CATLM = 0, 0, 1, 1,

The maximum number of drag coefficients that may be solved for is fifteen. The six Category 1 variables must be included in the solution vector whenever drag coefficients are solved for. If the position and velocity are known sufficiently, further change during the differential correction process can be prevented by imposing small bounds. It should be noted that bounds must be specified for the drag solution variables; the bounds for the drag variables are entered on the BNDS card immediately following the state variables and preceding any Category 2 variables.

3.3.3 Steering Ephemeris

The steering ephemeris option is flagged by setting Column 54 of the JDC card. The radar parameters are computed from the reference trajectory and output for each sensor in the current sensor table. Two input variables are provided for the control of horizon limits and output units.

Variable Name	Description
ECRIT	The ECRIT card (critical elevation) specifies the elevation in degrees above which the steering ephemeris is printed. The nominal value is -5 degrees.
RDFLG	The RDFLG card is a flag for the output units of range rate and range acceleration in the steering ephemeris. If RDFLG = 0., the nominal setting, the output units of range rate and range acceleration are kilometers/minute and kilometers/(minute)^2 respectively; if RDFLG = 1., the units are kilometers/second and meters/(second)^2 respectively.

3.3.4 Linear Constraints

Two input arrays are required to specify a linear constraints case: BIJ and XIJ. The arrays define the constraint matrix, B, which is sparse. The set-up of the constraint matrix can be best explained by example.

Assume that there are n parameters to be solved for, $(X_1, X_2, \cdots X_n) = X$. The order of X corresponds to the order of the solution variables; that is, the Category 1 variables, the drag variables (CATLM), and the

Category 2 variables. Also assume that there are m linear constraints to be placed on these variables. If n = 8 and m = 3, and the constraints are as listed below:

$$X_3 = X_4 \ , \ 5X_5 = X_6 \ , \ X_7 = 2X_8$$

then the dimension of the constrained system is d = n - m = 5 for this example. Stating the problem in the functional form:

$$X = B\overline{X}$$

where $\overline{X}$ is the vector of constrained variables.

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} \overline{X}_1 \\ \overline{X}_2 \\ \overline{X}_3 \\ \overline{X}_4 \\ \overline{X}_5 \end{bmatrix}$$

Variable Name	Description
BIJ	Each entry of the BIJ array is separated by a comma in the NAMELIST convention and corresponds to a non-zero element. The i, j th element of the constraint matrix B is entered as 100i + j. For example, if i = 12, and j = 9 is a non-zero element, it is entered as 1209 in the BIJ array. For the constraint matrix given above, the BIJ card would look like the following: BIJ = 101, 202, 303, 403, 504, 604, 705, 805,
XIJ	The XIJ card is an array of numerical values of the non-zero elements of the constraint matrix B. The input sequence of the XIJ entries is in a one-to-one correspondence with the BIJ array. The XIJ card for the constraint matrix given above is as follows: XIJ = 1., 1., 1., 1., 1., 5., 1., 0.5,

3.4 SENSOR CARDS

The sensor card format of the NRTPD2 program is given on the following pages. There are six types of sensor cards, the first three of which are acceptable to the PREMOD and NRTPOD programs as described in Reference 1. The remaining three sensor cards have been added to provide the input for the functional standard deviation option.

The sensor identification is (up to) three alphanumerical characters entered in the first three columns. Two additional identifying columns are provided to permit biases to be different from pass to pass. The full identification parallels the format on the observation cards.

The type column indicates the type of information on the card according to the following key:

Type Number (Column 7)

Field	1 Locations	2 Biases	3 Standard Deviations
Field 4	Latitude	Azimuth bias	σ_A
Field 5	Longitude	Elevation bias	σ_E
Field 6	Height	Range bias	σ_R
Field 7		$\dot{R}$ bias	$\sigma_{\dot{R}}$
Field 8		$\ddot{R}$ bias	$\sigma_{\ddot{R}}$
Field 9		Time bias	
Field 10	Station name		

Field	4 Sensor Constants	5 θ Table	6 $f(\theta)$ Table
Field 4	B_A	θ_1	$f(\theta_1)$
Field 5	B_E	θ_2	$f(\theta_2)$
Field 6	B_R	θ_3	$f(\theta_3)$
Field 7	$B_{\dot{R}}$	θ_4	$f(\theta_4)$
Field 8	$B_{\ddot{R}}$	θ_5	$f(\theta_5)$
Field 9		θ_6	$f(\theta_6)$
Field 10		θ_7	$f(\theta_7)$

The data fields are each nine columns wide. Data may be input in any of the conventional FORTRAN arrangements, that is, with either implicit decimal point, and with or without a right adjusted exponent of ten preceded by a punched plus or minus sign. If the first column of a field is not punched -(minus), the contained value is assumed positive. The implicit decimal point is located between the first and second column of each field. On the card, implicit decimal points appear between the following pairs of columns: 9-10; 19-20; 29-30; 39-40; 49-50; 59-60; and 69-70.

Field	Columns	Description
1	1-3	Station identification
2	4-5	Pass number: applicable to Type 2 (biases) cards only
	6	Space = blank
3	7	Type
		Blank or 0: Error on input, disregarded
		1: Interpret ϕ, λ, h
		2: Interpret A_b, E_b, R_b, $\dot{R}_b$, $\ddot{R}_b$, t_b
		3: Interpret σ_A, σ_E, σ_R, $\sigma_{\dot{R}}$, $\sigma_{\ddot{R}}$
		4: Interpret B_A, B_E, B_R, $B_{\dot{R}}$, $B_{\ddot{R}}$
		5: Interpret θ_1, θ_2, θ_3, θ_4, θ_5, θ_6, θ_7
		6: Interpret $f(\theta_1)$, $f(\theta_2)$, $f(\theta_3)$, $f(\theta_4)$, $f(\theta_5)$, $f(\theta_6)$, $f(\theta_7)$,
	8	Space = blank
4	9-17	Type 1: Geodetic latitude; ϕ, degrees (positive north)
		Type 2: Bias in Azimuth; A_b, degrees
		Type 3: σ_A, degrees
		Type 4: B_A
		Type 5: θ_1, degrees
		Type 6: $f(\theta_1)$, m^2
	18	Space = blank
5	19-27	Type 1: Longitude; λ, degrees (positive east of Greenwich)
		Type 2: Bias in Elevation; E_b, degrees

Field	Columns	Description
		Type 3: σ_E, degrees
		Type 4: B_E
		Type 5: θ_2, degrees
		Type 6: $f(\theta_2)$, m^2
	28	Space = blank
6	29-37	Type 1: Height; h, meters (positive above ellipsoid)
		Type 2: Bias in Range; R_b, km
		Type 3: σ_R, km
		Type 4: B_R
		Type 5: θ_3, degrees
		Type 6: $f(\theta_3)$, m^2
	38	Space = blank
7	39-47	Type 1:
		Type 2: Bias in first time derivative of range; $\dot{R}_b$, km/sec
		Type 3: $\sigma_{\dot{R}}$, km/sec
		Type 4: $B_{\dot{R}}$
		Type 5: θ_4, degrees
		Type 6: $f(\theta_4)$, m^2
	48	Space = blank
8	49-57	Type 1:
		Type 2: Bias in second time derivative of range; $\ddot{R}_b$, km/sec/sec
		Type 3: $\sigma_{\ddot{R}}$, km/sec/sec
		Type 4: $B_{\ddot{R}}$
		Type 5: θ_5, degrees
		Type 6: $f(\theta_5)$, m^2
	58	Space = blank
9	59-67	Type 1:
		Type 2: Bias in assigned time of observation; t_b, sec
		Type 3:
		Type 4:
		Type 5: θ_6, degrees
		Type 6: $f(\theta_6)$, m^2

Field	Columns	Description
	68	Space = blank
10	69-77	Type 1: Station Name
		Type 2:
		Type 3:
		Type 4:
		Type 5: θ_7, degrees
		Type 6: $f(\theta_7)$, m^2
	78	Space = blank
11	79-80	Not used, to be punched with some unambiguous mnemonic to identify this card conveniently as a sensor card.

The sensor constants are in mixed units so that the term $B_i R^4 / f(\theta)$ is dimensionally consistent with σ^2, where R is in earth-radii and $f(\theta)$ in meters squared. See Section 2. 2 for a more complete discussion of the functional standard deviation option.

3. 5 OBSERVATION CARDS

The observation card format for the NRTPD2 program is the same as for the other Lincoln Laboratory orbit determination programs; the only difference is the program response. Since the NRTPOD program does not accept the $\ddot{R}$ observable, it will ignore these observations if input. See Section 1. 2. 6 of Reference 1 for a description of the observation cards.

3. 6 EPHEMERIS CARDS

The format and use of the ephemeris cards is unchanged.

4. PROGRAM OUTPUT

The printed output of the NRTPOD modifications is explained and supplemented with samples in this section. The format is generally the same as NRTPOD; in most instances, the formats have been extended from or added to the existing version. A complete output guide is given below; however, if the format is unchanged from NRTPOD, it is not treated in this section.

Data	Number of Pages	Section
JDC Print	1	4.1
Input Listing	1	4.2*
Run Header	1	4.3
Sensor Information	1	4.4
Observations	1 or more	4.5
Residuals	1 or more	4.6
Functional Standard Deviations	1 or more	4.7
Mean and RMS by Sensor and Type	1	4.8*
Iteration Summary	1	4.9
Trajectory and Steering	1 or more	4.10
Matrix Update	1 or more	4.11

4.1 JDC PRINT

The JDC Print is the first page of a given run. Across the top of the page is a facsimile card image of the JDC card. Below this line is a JDC column content itemization and a short description of the option which is called by that particular JDC flag. Figure 4-1 is an example of the JDC print page.

4.2 INPUT LISTING

See Section 2.3.2 of Reference 1.

*Sample output not included in this section. See Section 2.3 of Reference 1.

1.1.1.1.0.0.0.0.0.0.1.0.1.0.0.0.0.0.0.0.1.0.0.1.1.0.0.0.0.0.

JDC OPTIONS

CARD COLUMN	VALUE	DESCRIPTION
31	1	SENSORS AND OBS
32	1	PRINT SENSORS
33	1	PRINT OBS
34	1	PRINT FUNCTIONAL SIGMA INPUT
41	1	CURVE FIT DESIRED
43	1	FUNCTIONAL SIGMA PRINT
51	1	TRAJECTORY PROPAGATION DESIRED
54	1	STEERING EPHEMERIS WITH TRAJ PRINT
55	1	UPDATE DESIRED

Figure 4-1. Sample JDC Print

4.3 RUN HEADER

The run identification page remains essentially as in NRTPOD. The layered drag model, which is used in the integration of the trajectory, is listed under "DRAG MODEL." The selected value of the drag parameter ($C_D A/m$, m^2/kg) at a particular altitude (km) is listed. Figure 4-2 is a sample of the Run Header.

4.4 SENSOR INFORMATION

The sensor data associated with the functional standard deviation option can be printed on option. When Column 34 of the JDC card is flagged, a separate page titled "FUNCTIONAL SIGMA INPUT" is printed, following the Run Header. There are three lines of output for each sensor, representing the information contained on sensor card Types 4, 5, and 6 respectively. For each sensor having functional standard deviation input, the following is printed:

Line 1:	ID	B_R	B_{AZ}	B_{EL}	$B_{\dot{R}}$	$B_{\ddot{R}}$		
Line 2:	ID	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
Line 3:	ID	$f(\theta_1)$	$f(\theta_2)$	$f(\theta_3)$	$f(\theta_4)$	$f(\theta_5)$	$f(\theta_6)$	$f(\theta_7)$

where the B's are sensor dependent constants; θ, the aspect angle; and $f(\theta)$, the radar cross section. These quantities are defined in Section 2.2.1. The second column refers to the type of sensor card. Figure 4-3 is a sample output of functional standard deviation information by sensor.

4.5 OBSERVATIONS

The observations print has been expanded to include the $\ddot{R}$ observables. A "TYPE" column has also been added between the "ID" and "T-TO" columns for the purpose of identifying the observable type. The Type 0 (zero) observables are R, A, E, and $\dot{R}$. The Type 1 observable is $\ddot{R}$. This method of identifying observables is explained in Reference 1, Section 1.2.6. The units of $\ddot{R}$ as printed are meters/second/second. Figure 4-4 is a sample Observations Print.

4.6 RESIDUALS

The residuals print has been extended to include $\ddot{R}$ residuals. As in the observations print, a "TYPE" column has been added to signify the

ORBIT DETERMINATION PROGRAM - NRTPOD

VEHICLE NO. VEHICLE NAME

ALPHA G ZERO
73.581959

INITIAL CONDITIONS

YEAR	MONTH	DAY	HOUR	MINUTE	SECOND
65	12	5	17	7	-0.

X	Y	Z	XDOT	YDOT	ZDOT
1.7650249E 03	5.8703489E 03	-2.6603089E 03	-7.2957926E 00	1.2722993E 00	-2.1269383E 00

ALPHA	DELTA	BETA	AZIMUTH	R	V
7.3265659E 01	-2.3460134E 01	8.9721918E 01	1.0738603E 02	6.6823316E 03	7.7052712E 00

DRAG MODEL

ALTITUDE(KM)	CDA/M (MT**2/KGM)
1.00000000E 15	2.42938137E 01
3.14839995E 02	2.30938137E 01
2.96319997E 02	2.30938137E 01
9.99999988E-09	2.42938137E 01

NO RADIATION PRESSURE

SENSOR LOCATIONS

ID	LATITUDE	LONGITUDE	ALT	R BIAS	A BIAS	E BIAS	R. BIAS	R..BIAS	TIME BIAS
05	-28.0000	122.5000	5.	0.	0.	0.	0.	0.	0.
15	-26.2000	164.5000	15.	0.	0.	0.	0.	0.	0.
25	-12.2000	201.3000	25.	0.	0.	0.	0.	0.	0.

Figure 4-2. Sample Run Header

FUNCTIONAL SIGMA INPUT

ID	TYPE	B R THETA1 F(THETA1)	B AZ THETA2 F(THETA2)	B EL THETA3 F(THETA3)	B R. THETA4 F(THETA4)	B R.. THETA5 F(THETA5)	THETA6 F(THETA6)	THETA7 F(THETA7)
05	4	2.50000E-05	6.67000E-06	6.67000E-06	2.50000E-09	5.62500E-03		
	5	3.00000E 01	5.00000E 01	6.00000E 01	7.00000E 01	9.00000E 01	1.20000E 02	1.50000E 02
	6	3.00000E 00	7.00000E 00	9.00000E 00	1.05000E 01	1.08900E 01	6.20000E 00	5.95000E 00
15	4	3.50000E-05	7.67000E-06	7.67000E-06	3.50000E-09	6.62500E-03		
	5	3.00000E 01	6.00000E 01	7.00000E 01	8.00000E 01	9.00000E 01	1.00000E 02	1.20000E 02
	6	5.00000E 00	6.00000E 00	7.00000E 00	8.00000E 00	5.00000E 00	4.00000E 00	3.00000E 00
25	4	4.50000E-05	8.67000E-06	8.67000E-06	4.50000E-09	7.62500E-03		
	5	5.00000E 01	7.00000E 01	8.00000E 01	9.00000E 01	1.10000E 02	1.30000E 02	1.50000E 02
	6	2.00000E 01	2.40000E 01	3.50000E 01	4.80000E 01	4.10000E 01	3.90000E 01	1.20000E 01

Figure 4-3. Sample Functional Sigma Input Page

ID	TYPE	T-TO	YR	MN	DY	HR	MIN	SECS	R SIGMA R	AZ SIGMA A	EL SIGMA E	RDOT SIGMA R.	R.. SIGMA R..
									OBSERVATION TYPE				
05	0	0.400	65	12	5	17	7	23.998	1983.7720	278.8611	0.2217	-6.9296	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	0.400	65	12	5	17	7	23.998	0.	0.	0.	0.	0.115255E-00
									0.	0.	0.	0.	10.000000E-03
05	0	0.600	65	12	5	17	7	35.996	1900.6350	278.8350	1.0100	-6.9276	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	0.600	65	12	5	17	7	35.996	0.	0.	0.	0.	0.228848E-00
									0.	0.	0.	0.	10.000000E-03
05	0	0.800	65	12	5	17	7	47.995	1817.5310	278.8084	1.8360	-6.9240	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	0.800	65	12	5	17	7	47.995	0.	0.	0.	0.	0.359458E-00
									0.	0.	0.	0.	10.000000E-03
05	0	1.000	65	12	5	17	7	59.995	1734.4800	278.7813	2.7048	-6.9188	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	1.000	65	12	5	17	7	59.995	0.	0.	0.	0.	0.511309E 00
									0.	0.	0.	0.	10.000000E-03
05	0	1.200	65	12	5	17	8	11.992	1651.5020	278.7536	3.6231	-6.9117	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	1.200	65	12	5	17	8	11.992	0.	0.	0.	0.	0.689934E 00
									0.	0.	0.	0.	10.000000E-03
05	0	1.400	65	12	5	17	8	23.992	1568.6240	278.7252	4.5987	-6.9022	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	1.400	65	12	5	17	8	23.992	0.	0.	0.	0.	0.902679E 00
									0.	0.	0.	0.	10.000000E-03
05	0	1.600	65	12	5	17	8	35.991	1485.8770	278.6960	5.6409	-6.8898	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	1.600	65	12	5	17	8	35.991	0.	0.	0.	0.	0.115940E 01
									0.	0.	0.	0.	10.000000E-03
05	0	1.800	65	12	5	17	8	47.989	1403.2970	278.6658	6.7614	-6.8741	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	1.800	65	12	5	17	8	47.989	0.	0.	0.	0.	0.147352E 01
									0.	0.	0.	0.	10.000000E-03
05	0	2.000	65	12	5	17	8	59.989	1320.9300	278.6346	7.9748	-6.8542	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.
05	1	2.000	65	12	5	17	8	59.989	0.	0.	0.	0.	0.186363E 01
									0.	0.	0.	0.	10.000000E-03
05	0	2.200	65	12	5	17	9	11.987	1238.8330	278.6020	9.2992	-6.8290	0.
									9.9999998E-01	10.0000E-03	10.0000E-03	10.0000E-03	0.

Figure 4-4. Sample Observations Print

type of observable in each line of output. The units of $\ddot{R}$ residuals are meters/second/second. Figure 4-5 is a sample Residuals Print.

4.7 FUNCTIONAL STANDARD DEVIATIONS

When the functional standard deviations option is used, the computed standard deviation as applied to each residual can be printed along with the residual output for each iteration of the differential correction. For each page of residuals, there is a corresponding functional standard deviations page. Both pages have the residual number (N), hence it is easy to match a functional standard deviation with its particular residual and observation. Figure 4-6 is a sample Functional Standard Deviations Print. The column symbols and their description are as follows:

Column Symbol	Description
ID	Observing station's identification
TYPE	Type of observable 0: RAE$\dot{R}$ 1: $\ddot{R}$
TIME (MIN)	Time in minutes from epoch
N	Serial number assigned to each residual for identification purposes. It is constant through the run.
SIGMA R (KM)	Computed standard deviation of range in kilometers
SIGMA A (DEG)	Computed standard deviation of azimuth in degrees
SIGMA E (DEG)	Computed standard deviation of elevation in degrees
SIGMA R. (KM/SEC)	Computed standard deviation of range rate in kilometers/second
SIGMA R.. (MT/SEC**2)	Computed standard deviation of range acceleration in meters/second/second

4.8 MEAN AND STANDARD DEVIATIONS BY SENSOR

The mean and standard deviation of $\ddot{R}$ data has been added to the R, A, E, $\dot{R}$ list. See Section 2.3.5 of Reference 1.

RESIDUALS PRINT

IO	T	TIME (MIN)	N	R (KM)	A (DEG)	E (DEG)	RDOT (KM/SEC)	R.. (MT/S2)	U (KM)	V (KM)	W (KM)	VM (KM)	BETA (DEG)
05	1	7.399	180	0.	0.	0.	0.	-0.000136	-0.0001	0.0000	-0.0000	0.0001	0.
05	0	7.599	181	-0.05628	-0.00	0.00	-0.000273	0.	-0.0201	-0.0526	0.0013	0.0563	0.000
05	1	7.599	185	0.	0.	0.	0.	-0.000302	-0.0001	-0.0000	0.0000	0.0001	0.
05	0	7.799	186	-0.05280	0.00	-0.00	-0.000179	0.	-0.0223	-0.0481	-0.0002	0.0530	-0.000
05	1	7.799	190	0.	0.	0.	0.	-0.000239	-0.0001	-0.0000	0.0000	0.0001	0.000
05	0	7.999	191	-0.06129	0.00	-0.00	-0.000185	0.	-0.0248	-0.0563	-0.0001	0.0615	-0.000
05	1	7.999	195	0.	0.	0.	0.	-0.000227	-0.0001	-0.0000	0.0000	0.0001	0.000
05	0	8.199	196	-0.05842	-0.00	-0.00	-0.000203	0.	-0.0255	-0.0530	0.0016	0.0589	0.000
05	1	8.199	200	0.	0.	0.	0.	-0.000284	-0.0001	-0.0000	0.0000	0.0001	-0.000
05	0	8.399	201	-0.06047	0.00	-0.00	-0.000253	0.	-0.0267	-0.0548	-0.0004	0.0610	-0.000
05	1	8.399	205	0.	0.	0.	0.	-0.000270	-0.0001	-0.0000	0.0000	0.0001	-0.000
05	0	8.599	206	-0.07074	-0.00	-0.00	-0.000273	0.	-0.0298	-0.0647	0.0012	0.0712	0.000
05	1	8.599	210	0.	0.	0.	0.	-0.000255	-0.0001	-0.0000	0.0000	0.0001	0.
05	0	8.799	211	-0.06665	0.00	-0.00	-0.000293	0.	-0.0310	-0.0600	0.0006	0.0675	0.000
05	1	8.799	215	0.	0.	0.	0.	-0.000269	-0.0001	-0.0000	0.0000	0.0001	-0.000
05	0	8.999	216	-0.07620	0.00	-0.00	-0.000203	0.	-0.0331	-0.0694	-0.0004	0.0769	-0.000
05	1	8.999	220	0.	0.	0.	0.	-0.000251	-0.0001	-0.0000	0.0000	0.0001	0.
05	0	9.199	221	-0.07865	-0.00	-0.00	-0.000283	0.	-0.0363	-0.0710	0.0019	0.0797	0.000
05	1	9.199	225	0.	0.	0.	0.	-0.000244	-0.0001	-0.0000	0.0000	0.0001	0.000
05	0	9.399	226	-0.07561	-0.00	-0.00	-0.000232	0.	-0.0393	-0.0668	0.0019	0.0775	0.000
05	1	9.399	230	0.	0.	0.	0.	-0.000260	-0.0001	-0.0000	0.0000	0.0001	0.
05	0	9.599	231	-0.08551	0.00	-0.00	-0.000287	0.	-0.0397	-0.0771	0.0006	0.0867	0.000
05	1	9.599	235	0.	0.	0.	0.	-0.000249	-0.0001	-0.0000	0.0000	0.0001	0.000
05	0	9.799	236	-0.08813	0.00	-0.00	-0.000241	0.	-0.0416	-0.0793	-0.0004	0.0895	-0.000
05	1	9.799	240	0.	0.	0.	0.	-0.000235	-0.0001	-0.0000	0.0001	0.0001	0.000
15	0	10.399	241	0.05950	0.00	-0.00	0.000169	0.	-0.0519	-0.0792	0.0022	0.0947	0.000
15	1	10.399	245	0.	0.	0.	0.	-0.000589	-0.0001	0.0000	0.0001	0.0001	0.000
15	0	10.599	246	0.06128	0.00	-0.00	0.000097	0.	-0.0542	-0.0818	0.0008	0.0982	0.000
15	1	10.599	250	0.	0.	0.	0.	-0.000719	-0.0000	0.0000	0.0000	0.0001	0.000
15	0	10.799	251	0.06826	0.00	-0.00	0.000080	0.	-0.0579	-0.0905	0.0005	0.1074	0.000
15	1	10.799	255	0.	0.	0.	0.	-0.000894	-0.0001	0.0000	0.0000	0.0001	0.000
15	0	10.999	256	0.06356	0.00	-0.00	0.000094	0.	-0.0603	-0.0865	0.0019	0.1055	0.000
15	1	10.999	260	0.	0.	0.	0.	-0.001105	-0.0001	0.0000	0.0000	0.0001	0.
15	0	11.199	261	0.07140	0.00	-0.00	0.000053	0.	-0.0611	-0.0953	0.0019	0.1132	0.000
15	1	11.199	265	0.	0.	0.	0.	-0.001137	-0.0001	0.0000	0.0000	0.0001	0.
15	0	11.399	266	0.07171	0.00	-0.00	0.000101	0.	-0.0659	-0.0976	0.0009	0.1177	0.000

Figure 4-5. Sample Residuals Print

FUNCTIONAL SIGMAS

ID	TYPE	TIME (MIN)	N	SIGMA R (KM)	SIGMA A (DEG)	SIGMA E (DEG)	SIGMA R. (KM/SEC)	SIGMA R.. (MT/SEC**2)
05	1	7.399	180	0.	0.	0.	0.	0.10071360E-01
05	0	7.599	181	0.99999999E 00	0.10203473E-01	0.10000113E-01	0.99999994E-02	0.
05	1	7.599	185	0.	0.	0.	0.	0.10095711E-01
05	0	7.799	186	0.99999999E 00	0.10153578E-01	0.10000149E-01	0.99999994E-02	0.
05	1	7.799	190	0.	0.	0.	0.	0.10125941E-01
05	0	7.999	191	0.99999999E 00	0.10114624E-01	0.10000194E-01	0.99999995E-02	0.
05	1	7.999	195	0.	0.	0.	0.	0.10162917E-01
05	0	8.199	196	0.99999999E 00	0.10084129E-01	0.10000248E-01	0.99999995E-02	0.
05	1	8.199	200	0.	0.	0.	0.	0.10207524E-01
05	0	8.399	201	0.99999999E 00	0.10060291E-01	0.10000312E-01	0.99999995E-02	0.
05	1	8.399	205	0.	0.	0.	0.	0.10260701E-01
05	0	8.599	206	0.09999999E 01	0.10041790E-01	0.10000389E-01	0.99999998E-02	0.
05	1	8.599	210	0.	0.	0.	0.	0.10323402E-01
05	0	8.799	211	0.09999999E 01	0.10027644E-01	0.10000478E-01	0.99999998E-02	0.
05	1	8.799	215	0.	0.	0.	0.	0.10396571E-01
05	0	8.999	216	0.09999999E 01	0.10017101E-01	0.10000584E-01	0.99999998E-02	0.
05	1	8.999	220	0.	0.	0.	0.	0.10481197E-01
05	0	9.199	221	0.09999999E 01	0.10009588E-01	0.10000705E-01	0.99999999E-02	0.
05	1	9.199	225	0.	0.	0.	0.	0.10578218E-01
05	0	9.399	226	0.09999999E 01	0.10004655E-01	0.10000844E-01	0.09999999E-01	0.
05	1	9.399	230	0.	0.	0.	0.	0.10688566E-01
05	0	9.599	231	0.09999999E 01	0.10001947E-01	0.10001003E-01	0.09999999E-01	0.
05	1	9.599	235	0.	0.	0.	0.	0.10813170E-01
05	0	9.799	236	0.09999999E 01	0.10001184E-01	0.10001183E-01	0.09999999E-01	0.
05	1	9.799	240	0.	0.	0.	0.	0.10952863E-01
15	0	10.399	241	0.10000000E 01	0.10001759E-01	0.10001199E-01	0.09999999E-01	0.
15	1	10.399	245	0.	0.	0.	0.	0.10987504E-01
15	0	10.599	246	0.09999999E 01	0.10003966E-01	0.10001010E-01	0.09999999E-01	0.
15	1	10.599	250	0.	0.	0.	0.	0.10838568E-01
15	0	10.799	251	0.09999999E 01	0.10008333E-01	0.10000846E-01	0.09999999E-01	0.
15	1	10.799	255	0.	0.	0.	0.	0.10706201E-01
15	0	10.999	256	0.09999999E 01	0.10015193E-01	0.10000702E-01	0.09999999E-01	0.
15	1	10.999	260	0.	0.	C.	0.	0.10589460E-01
15	0	11.199	261	0.09999999E 01	0.10024972E-01	0.10000577E-01	0.99999999E-02	0.
15	1	11.199	265	0.	0.	0.	0.	0.10487288E-01
15	0	11.399	266	0.09999999E 01	0.10038212E-01	0.10000470E-01	0.99999998E-02	0.

Figure 4-6. Sample Functional Standard Deviations Print

4.9 ITERATION SUMMARY

This page shows the results and convergence status for a given iteration. The format and content of this page remains unchanged except that the Category 1 list has been extended to include the altitude dependent drag parameters. The drag parameters are labeled "LAMBDA 1, LAMBDA 2, . . . LAMBDA N, where LAMBDA 1 is the uppermost layer and LAMBDA N the lowest. The maximum number of layers which may be included in the solution vector is fifteen. Figure 4-7 is a sample iteration summary of a determination including the satellite position and velocity and drag parameters at nine altitudes in the solution vector. The iteration summary includes the normal matrix, the variance-covariance matrix, and the correlation matrix. Since the solution vector is of dimension 15, the matrix print of the three aformentioned matrices does not appear entirely on the first page. Hence, only part of the correlation matrix appears in Figure 4-7, the first page of the iteration summary.

4.10 TRAJECTORY AND STEERING

The trajectory output has been extended to include the Kozai mean elements at each print (DELTT) time. Immediately following the standard trajectory print, the following mean elements are output:

Symbol	Definition	
A	Mean semi-major axis	- earth-radii
E	Mean eccentricity	- N. D.
I	Mean orbital inclination	- degrees
NODE	Mean right ascension of the ascenting node	- degrees
OM	Mean argument of perihelion	- degrees
M	Mean mean anomaly	- degrees
NDOT	Rate of change of right ascension of ascending node	- degrees/day
ODOT	Rate of change of argument of perihelion	- degrees/day

ITERATION NUMBER 1

CATEGORY 1 VARIABLES		DELTA	OLD	NEW	SIGMA	BOUNDS
1	ALPHA	-0.22341133E-06	0.14667960E 03	0.14667960E 03	0.44523887E-04	0.99999999E 00
2	DELTA	-0.37836007E-05	0.10725830E 02	0.10725827E 02	0.74620796E-04	0.99999999E 00
3	BETA	-0.35990185E-04	0.11195949E 03	0.11195945E 03	0.12248758E-02	0.99999999E 00
4	AZ	0.21590819E-03	0.23639420E 03	0.23639441E 03	0.34519801E-02	0.99999999E 00
5	R	-0.19343250E-03	0.64998343E 04	0.64998340E 04	0.34371310E-02	0.49999999E 01
6	V	0.34548036E-04	0.64700545E 01	0.64700890E 01	0.81705334E-03	0.20000000E-00
7	LAMBDA 1	-0.60666882E-04	0.39999998E-02	0.39393330E-02	0.29211566E-00	0.39999998E-02
8	LAMBDA 2	0.34860163E-03	0.39999998E-02	0.43486015E-02	0.15189849E-01	0.39999998E-02
9	LAMBDA 3	-0.39789823E-03	0.19999999E-02	0.16021017E-02	0.18269486E-02	0.19999999E-02
10	LAMBDA 4	-0.31046887E-04	0.20000000E-03	0.16895311E-03	0.27952631E-03	0.20000000E-03
11	LAMBDA 5	-0.37571841E-04	0.99999998E-04	0.62428156E-04	0.68232170E-04	0.99999998E-04
12	LAMBDA 6	0.36125451E-04	0.59999999E-04	0.96125451E-04	0.20285680E-04	0.59999999E-04
13	LAMBDA 7	0.20552998E-04	0.49999999E-04	0.70552997E-04	0.64225130E-05	0.49999999E-04
14	LAMBDA 8	0.10146292E-04	0.39999999E-04	0.50146291E-04	0.28184474E-05	0.39999999E-04
15	LAMBDA 9	-0.51586498E-05	0.39999999E-04	0.34841349E-04	0.27083981E-04	0.39999999E-04

SOLUTION IS CONVERGING

SOLUTION IS AFFECTED BY BOUNDS

CURRENT RMS	1.060577
PREDICTED RMS	0.971989
BEST RMS	1.060577

MEAN ELEMENTS FROM NEW

A	1.9886610D 00(ER)	NODE	2.6514109D 02(DEG)	NDOT	-2.1285507D-02(D/DAY)
E	1.1510486D-02	OM	2.1843268D 02(DEG)	ODOT	-4.4796980D-01(D/DAY)
I	8.8642613D 01(DEG)	M	1.1517518D 02(DEG)		

CORRELATION MATRIX

	1	2	3	4	5	6
1	1.000000000					
2	0.665326603	1.000000000				
3	0.153708242	-0.209599964	1.000000000			
4	0.802833229	0.874134786	0.003177181	1.000000000		
5	-0.041396491	-0.032528795	0.811555073	0.051357932	1.000000000	
6	0.304581948	-0.414556548	0.173853397	-0.199712731	-0.302621901	1.000000000
7	0.319818392	-0.403772309	0.174713984	-0.183505602	-0.304481454	0.998555787

Figure 4-7. Sample Iteration Summary Print (complete output not shown)

Figure 4-8 is a sample output of the trajectory print with mean elements.

The steering ephemeris appears in the trajectory and update print section, and is printed at each update time (Δt, t). Following the trajectory block, if it is requested, the following output constitutes the steering ephemeris:

First Block

The first block contains atmospheric parameters only (which are sensor independent); hence, the following quantities are printed once per update time.

Column Symbol	Description
HEIGHT (KM)	Altitude of spacecraft above reference ellipsoid in kilometers
DENSITY (KG/M**3)	Density of the atmosphere at the altitude given in the previous column in kilograms/(meters)^3
CDAM (M**2/KG)	C_DA/M, Ballistic coefficient at current altitude in (meters)^2/kilogram

Second Block

In the second block of the steering output, there is one line of print for each sensor.

Column Symbol	Description
STATION	Sensor identification tag
AZIMUTH (DEG)	Topocentric azimuth of vehicle from north, in degrees.
ELEVATION (DEG)	Topocentric elevation of vehicle from horizon, in degrees.
RANGE (KM)	Topocentric range of vehicle in kilometers
RDOT (KM/MIN) or RDOT (KM/SEC)	Range rate of vehicle in kilometers/minute or kilometers/second

```
18    MARCH  1966   20 HR    8 MIN   42.96 SEC   MINUTES FROM EPOCH    2640.00000   DAYS FROM OHR JAN   77.839386

   X  0.55723685D 03 XDOT -0.29451953D 01 ALFA   0.88450202D 02    AZ  0.60309910D 02  ALT   0.79787180D 04
   Y  0.20595935D 05 YDOT -0.42281977D 00 DLTA   0.13096049D 02     R  0.21153644D 05  LAT   0.13181226D 02
   Z  0.47930805D 04 ZDOT  0.15998932D 01 BETA   0.92150124D 02     V  0.33782555D 01  LON   0.33033979D 03

SMA   0.15171091D 05 NODE  0.66777132D 02    UX  0.26342357D-01 RPVX -0.12342085D 01 ALAT  0.25158330D 02
ECC   0.39584337D 00  OMG  0.20186981D 03    UY  0.97363530D 00 RPVY -0.12561524D 00  TAU  0.26520303D 03
INC   0.32207518D 02    M  0.18697218D 03    UZ  0.22658415D 00 RPVZ  0.68325781D 00  PRD  0.30994422D 03
1/A   0.42041565D 00    D -0.53173904D-01 APOG   0.79904457D 04  PRG  0.15051575D 04             ELLIPSE

                                             MEAN ELEMENTS

                 A  2.3782096D 00(ER )  NODE  6.6648781D 01(DEG)  NDOT -5.7135058D-01(D/DAY)
                 E  3.9596613D-01         OM  2.0206438D 02(DEG)  ODOT  8.7107223D-01(D/DAY)
                 I  3.2206016D 01(DEG)     M  1.8696919D 02(DEG)

18    MARCH  1966   21 HR    8 MIN   42.96 SEC   MINUTES FROM EPOCH    2700.00000   DAYS FROM OHR JAN   77.881053

   X -0.91283413D 04 XDOT -0.20821415D 01 ALFA   0.12418008D 03    AZ  0.73318956D 02  ALT   0.64908738D 04
   Y  0.13441996D 05 YDOT -0.35532473D 01 DLTA   0.27952950D 02     R  0.18394544D 05  LAT   0.28112577D 02
   Z  0.86223752D 04 ZDOT  0.32235559D 00 BETA   0.10999011D 03     V  0.41309554D 01  LON   0.35102860D 03

SMA   0.15170732D 05 NODE  0.66769496D 02    UX -0.49625267D 00 RPVX -0.10152574D 01 ALAT  0.61590085D 02
ECC   0.39591230D 00  OMG  0.20187188D 03    UY  0.73075998D 00 RPVY -0.91979267D 00  TAU  0.20517419D 03
INC   0.32203569D 02    M  0.25667766D 03    UZ  0.46874635D 00 RPVZ  0.35909285D 00  PRD  0.30993323D 03
1/A   0.42042559D 00    D -0.51518947D 00 APOG   0.79907400D 04  PRG  0.15044758D 04             ELLIPSE

                                             MEAN ELEMENTS

                 A  2.3782086D 00(ER )  NODE  6.6625021D 01(DEG)  NDOT -5.7135272D-01(D/DAY)
                 E  3.9596741D-01         OM  2.0210056D 02(DEG)  ODOT  8.7107551D-01(D/DAY)
                 I  3.2206016D 01(DEG)     M  2.5666565D 02(DEG)
```

Figure 4-8. Sample Trajectory Print

```
START TRAJECTORY
  END TRAJECTORY
                NODE RATE                         PERIGEE RATE                       ANOMALISTIC PER.
               4.6545908E 01                      6.6425353E 01                        5.5784037E 01

22 SEPTEMBER 1964  13 HR  38 MIN   4.50 SEC  MINUTES FROM EPOCH       0.       DAYS FROM OHR JAN 265.568138

   X  0.61641339E 04 XDOT -0.12843174E 01 ALFA  0.15611518E 02   AZ  0.23603176E 03  ALT  0.69038081E 02
   Y  0.17223930E 04 YDOT -0.51255840E 01 DLTA  0.10312445E 02    R  0.65053348E 04  LAT  0.10380431E 02
   Z  0.11645585E 04 ZDOT -0.36197091E 01 BETA  0.12020212E 03    V  0.64049494E 01  LON  0.16965041E 03

 SMA  0.48894461E 04 NODE  0.18073077E 03   UX  0.94755059E 00 RPVX  0.22819629E-00 ALAT  0.16196181E 03
 ECC  0.57848269E 00  OMG  0.31175105E 03   UY  0.26476623E-00 RPVY -0.55123092E 00  TAU  0.17191595E 02
 INC  0.14468142E 03    M  0.26236256E 03   UZ  0.17901593E-00 RPVZ -0.39259186E-00  PRD  0.56708445E 02
 1/A  0.13044755E 01    C -0.41569997E-00 APOG  0.72340238E 03  PRG -0.23310901E 04          ELLIPSE
```

HEIGHT (KM)	DENSITY (KG/M**3)	CDAM (M**2/ KG)
0.12785852E 03	0.82956296E-08	0.20485266E-02

STATION	AZIMUTH (DEG.)	ELEVATION (DEG.)	RANGE (KM)	RDOT (KM/MIN)	RDDOT (KM/MIN**2)
TA	65.296174	24.671696	0.29321734E 03	-0.40227349E 03	-0.82043088E 01
TB	85.352126	48.048399	0.17057622E 03	-0.35390702E 03	0.19994877E 03
TC	129.080631	40.760778	0.19330124E 03	-0.21697094E 03	0.58048771E 03
TD	189.090475	53.627551	0.15795253E 03	-0.23282464E 02	0.10041345E 04
TE	184.361622	32.699956	0.23124857E 03	0.61966093E 02	0.67105278E 03
TF	216.484110	27.117971	0.27067158E 03	0.19238749E 03	0.44991525E 03
TG	205.887253	20.018548	0.34919785E 03	0.19942525E 03	0.34031641E 03
TH	222.712980	16.086232	0.41677552E 03	0.25635415E 03	0.22211976E 03
TI	214.476610	12.917997	0.49233137E 03	0.25174885E 03	0.19229797E 03

Figure 4-9. Sample Trajectory plus Steering Ephemeris Output

RDDOT (KM/MIN**2) or RDDOT (MT/SEC**2)	Range acceleration of vehicle in kilometers/minute/minute or meters/second/second

Figure 4-9 is an example of a trajectory plus steering ephemeris print. The entire output given is for a single point in time, as printed on the first line of output. In this particular example, the steering ephemeris is printed for nine sensors; the locations of the sensors which are included in the steering ephemeris are normally printed on the Header Page.

4.11 MATRIX UPDATE

The matrix update output now includes eigenvalues, associated eigenvectors, and determinants as well as the normal matrix and a "sigma and rho" matrix. At each update time, immediately following the normal matrix, the following quantities are output:

a) Eigenvalues and Eigenvectors. The uppermost elements of the six (6) columns of print are the square roots of the eigenvalues of the position partition (upper 3 x 3) and velocity partition (lower 3 x 3) of the UVW covariance matrix. The three components of the associated normalized eigenvector are printed below the respective eigenvalue.

b) Principal Axis Alignment. The ordered yaw-roll-pitch rotation to align the UVW system with the principal axes of the error ellipsoid are printed for both the position and velocity partitions of the UVW covariance matrix. The positive rotation of yaw-roll-pitch are left, clockwise, and down respectively.

c) Determinants. The square roots of the determinants of the following (sub) matrices are printed:

   1) The position partition (upper 3 x 3) of the Cartesian covariance matrix

   2) The velocity partition (lower 3 x 3) of the Cartesian covariance matrix

   3) The 6 x 6 Cartesian covariance matrix

Figure 4-10 (two pages) is a sample matrix update for one update time. The matrix update example also includes steering ephemeris output which is printed between the trajectory block and the matrix update output. It should be noted that the output units of range rate and range acceleration

```
 5 DECEMBER  1965  17 HR  27 MIN  -0.   SEC  MINUTES FROM EPOCH     20.00000  DAYS FROM CHR JAN 339.727081

  X -0.58835667E 04 XDOT -0.33460053E 01 ALFA  0.15972569E 03   AZ  0.68854349E 02  ALT  0.16556294E 03
  Y  0.21733983E 04 YDOT -0.64126303E 01 DLTA -0.20176806E 02    R  0.66822325E 04  LAT -0.20301703E 02
  Z -0.23048241E 04 ZDOT  0.26196849E 01 BETA  0.90321745E 02    V  0.76928753E 01  LON  0.18367711E 03

SMA  0.66298980E 04 NODE  0.20145100E 03   UX -0.88047917E 00 RPVX -0.44847576E-00 ALAT  0.31447058E 03
ECC  0.96872065E-02  OMG  0.99364385E 02   UY  0.32525032E-00 RPVY -0.84798250E 00  TAU  0.11437222E 02
INC  0.28903949E 02    M  0.21574840E 03   UZ -0.34491826E-00 RPVZ  0.34520348E-00  PRD  0.89540292E 02
1/A  0.96203063E 00    D -0.57250340E-02 APOG  0.17060374E 03  PRG  0.10124613E 03            ELLIPSE


      HEIGHT           DENSITY               CDAM
       (KM)           (KG/M**3)           (M**2/ KG)
      ------          ---------           ----------
   0.30662256E 03   0.20394284E-10      0.24086063E 02


STATION       AZIMUTH       ELEVATION         RANGE              RDOT              RDDOT
               (DEG.)        (DEG.)            (KM)            (KM/SEC)         (MT/SEC**2)
              -------       ---------          -----            --------         -----------
  05         96.572600     -25.357612     0.61212540E 04     0.62382565E 01    -0.17866691E 01
  15         75.558893      -1.103245     0.21279551E 04     0.69184770E 01    -0.72576000E-01
  25        242.267208      -1.264774     0.21458639E 04    -0.69473498E 01    -0.26213283E-00

            SIGMA AND RHO MATRIX POLAR SPHERICAL COORDINATES

               1                2                3                4                5                6
  1  0.12436266E-03
  2  0.49846038E-02  0.11749514E-03
  3 -0.53878807E 00 -0.21400638E-00  0.20806196E-03
  4 -0.12715320E-00 -0.36017149E-00  0.15493370E-00  0.16227322E-03
  5 -0.16005352E-00 -0.69819763E-01 -0.48822782E-00  0.38990713E-01  0.94799841E-02
  6  0.48916135E-00  0.19424981E-00 -0.74545163E 00 -0.13499853E-00  0.48295514E-02  0.32821762E-04
  7 -0.48775520E-00 -0.20397425E-00  0.71970026E 00  0.14182466E-00  0.65152545E-01 -0.93517394E 00
  8  0.77385216E 00  0.32689624E-00 -0.59278587E 00 -0.21934541E-00 -0.26283604E-00  0.68568891E 00

               7                8
```

Figure 4-10. Sample Printout of a Single Trajectory-Steering-Matrix Update Output Block

```
7  0.27013164E-00
8 -0.77473164E 00  0.11391812E 01

        NORMAL MATRIX POLAR SPHERICAL COORDINATES

         1              2              3              4              5              6
1  0.36342783E 09
2  0.13026765E 09  0.13768023E 09
3  0.15889848E 09  0.65239861E 08  0.27614056E 09
4  0.60944040E 07  0.21563090E 08 -0.13683260E 07  0.44456534E 08
5  0.17655420E 07  0.71863795E 06  0.35197375E 07 -0.10597323E 05  0.57972430E 05
6 -0.44337454E 09 -0.17841548E 09  0.32114650E 09 -0.69777629E 07  0.33367630E 07  0.96274466E 10
7 -0.16620645E 06 -0.69142073E 05 -0.80358336E 05 -0.36175053E 03 -0.91801873E 03  0.11276838E 07
8 -0.35613564E 05 -0.15515512E 05  0.82222328E 03  0.47241709E 02  0.99588895E 02  0.10229062E 06

         7              8
7  0.22696743E 03
8  0.25070845E 02  0.71955980E 01

              POSITION                                                   VELOCITY

                              SQUARE ROOTS OF THE EIGENVALUES

0.92123412E-02  0.13850620E-01  0.13647936E-01     0.17321800E-04  0.36008557E-04  0.21783436E-04

                     ASSOCIATED EIGENVECTORS OF THE UVW COVARIANCE MATRIX

0.97622900E 00  -0.20727696E-00  0.63349968E-01     0.88570171E 00  0.46394157E-00  0.17049901E-01
0.21664581E-00   0.92449434E 00 -0.31364799E-00    -0.46365653E-00  0.88582748E 00 -0.18229607E-01
0.64452989E-02   0.31991635E-00  0.94742373E 00    -0.23560698E-01  0.82406808E-02  0.99968845E 00

      TO ALIGN U,V,W WITH PRINCIPAL AXES                 TO ALIGN U.,V.,W. WITH PRINCIPAL AXES

         YAW LEFT          18.3173 DEG                      YAW LEFT           1.0447 DEG
         ROLL CLOCKWISE    -3.6321 DEG                      ROLL CLOCKWISE    -0.9769 DEG
         PITCH DOWN        11.9873 DEG                      PITCH DOWN       -27.6461 DEG

                   SQUARE ROOTS OF DETERMINANTS OF CARTESIAN COVARIANCE MATRIX

                    POSITION                VELOCITY                6X6

                 0.17414310E-05          0.13587050E-13          0.86811937E-20
```

Figure 4-10. Sample Printout of a Single Trajectory-Steering-Matrix Update Output Block (Continued)

differ from the units as shown in Figure 4-8. See Section 3.3 regarding input requirements for steering ephemeris output units.

The diagonalized covariance matrix output described above is a permanent change to the program and is always printed when a matrix update is requested.

5. PROGRAM FUNCTIONAL DESCRIPTION

This section contains the subroutine descriptions of the NRTPD2 program. In addition to the standard subroutine descriptions, a complete subroutine glossary with functional descriptions is presented in Section 5.2. Because many of the subroutines were modified only slightly, a section describing the logic changes only has been included.

5.1 NRTPD2 SUBROUTINE OVERLAY

Figure 5-1 is a subroutine breakdown of the NRTPD2 overlay structure, although the overlay structure in terms of the principal options of the program remains unchanged. From this figure it can be seen which subroutines are in core as a function of the particular option of the program which is being used. The NRTPD2 subroutine overlay is similar to a corresponding diagram in Section 5.1 of Reference 1; the new subroutine and the ones which are included in one version only are identified for comparison purposes.

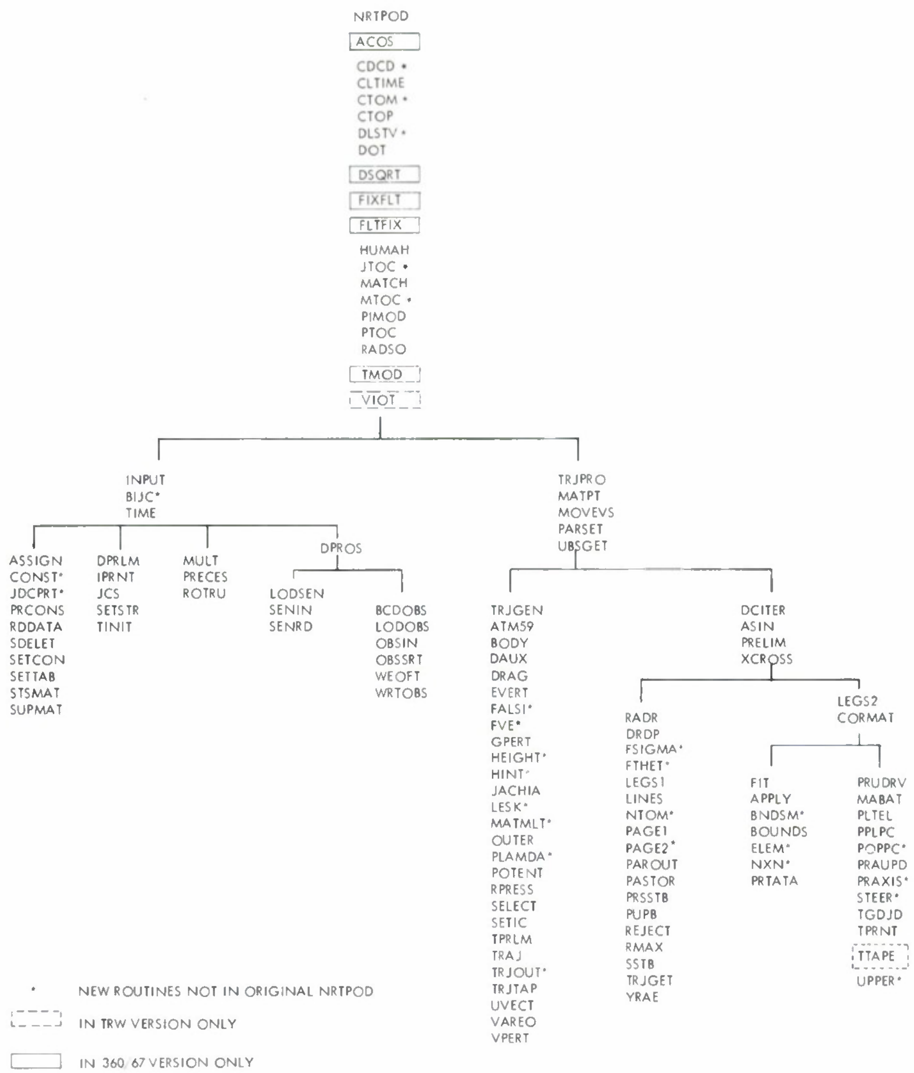


Figure 5-1. NRTPD2 Subroutine Overlay Structure

5.2 SUBROUTINE GLOSSARY

This section lists the subroutines which are used by the NRTPD2 program; that is, the list also includes the new subroutines which have been added as a result of the program modifications which this document describes.

Since many of the original subroutines remain unchanged, not all subroutine descriptions are given in this document. The documentation code which is explained below indicates how the program modifications affected the subroutine, and where the particular subroutine documentation is to be found. The following symbols constitute the documentation code:

M	Subroutine has been modified
N	New subroutine added as a result of new program options or system 360 conversion
U	Subroutine is unchanged
*	Subroutine modifications or descriptions given in Section 5.3.
( )	Reference document in which main subroutine documentation is to be found. The absence of ( ) indicates that the revised or new subroutine documentation is in Section 5.4.

Subroutine	Code	Functional Description
ACOS	N*	Arc cosine routine
APPLY	M*(1)	Applies correction to solution vector and prints iteration summary
ASIN	U(2)	Arc sine routine
ASSIGN	M	Establishes storage assignments for arrays in variable storage
ATM59	U(1)	Computes density of a static atmosphere (ARDC 1959 Model)
BCDOBS	M	Reads an observation card
BIJC	N	Sets up the linear constraints matrix in variable storage

Subroutine	Code	Functional Description
BNDSM	N	Sets up an appropriate set of bounds for the constrained solution vector
BODY	M	Computes perturbative acceleration due to sun and moon
BOUNDS	U(2)	Scales bounds with given scale factor
CDCD	N(1)	Modulates the input time
CLTIME	U(1)	Computes Gregorian time
CONST	N	Computes constants which are functions of input constants
CORMAT	U(1)	Computes correlation matrix
CTOM	N	Converts a Cartesian state vector to mean elements
CTOP	U(2)	Converts a Cartesian state vector to polar spherical coordinates (ADBARV)
DAUX	M	Driver for evaluating acceleration in integration
DCITER	U(1)	Driver for the computation of the normal matrix and one iteration of the DC
DLSTV	N	Computes the differentials for the MTOC and CTOM conversions
DOT	U(1)	Computes scalar product
DPRLM	M	Data initializing (partial)
DPROS	M	Driver for loading observation and sensor cards
DRAG	M	Computes drag perturbations
DRDP	M	Computes partials of observations w.r.t. category 1 variables
DSQRT	N*	Double precision square root routine
ELEM	N	Accesses the i, j element of a matrix stored in triangular form
EVERT	U(1)	Interpolates position of sun and moon

Subroutine	Code	Functional Description
FALSI	N	Determines altitude cutoffs
FIT	M	Logic control for DC options
FIXFLT	N*	Stores integer in floating array
FLTFIX	N*	Fixes a floating argument and stores it in a floating array
FSIGMA	N	Computes the functional standard deviation
FTHET	N	Interpolates the radar cross section from an input table
FVE	N	Determines the flags which indicate the ballistic coefficients which are to be solved for
GPERT	U(1)	Computes acceleration due to Earth's potential
HEIGHT	N	Computes the altitude of a vehicle above the Earth
HINT	N	Defines the two drag layers which bound the vehicle at a given time
HUMAH	U(1)	Converts a vector, A^TA matrix, or $(A^TA)^{-1}$ from machine units to human units or vice-versa
INPUT	M	Main driver for input processor
IPRNT	M*(1)	Prints header page
JACHIA	U(1)	Computes air density using Lockheed-Jacchia atmospheric model
JCS	U(1)	Sets up vector of zonal coefficients $J_2 \ldots J_{12}$ and two matrices of C's and S's for GPERT
JDCPRT	N	Prints the JDC card and describes the JDC options which have been flagged
JTOC	N(1)	Converts Julian date to calendar date
LEGS1	U(1)	Forms A^TA and A^TB given A and B
LEGS2	M	Least squares package solves AX = B

Subroutine	Code	Functional Description
LESK	N	Complex linear equation solver
LINES	M*(1)	Ejects page and prints heading at top of page
LODOBS	M*(1)	Main control for observation card processor
LODSEN	M*(1)	Main control for sensor card processor
MABAT	U(1)	Multiplies ABA^T where B is a lower triangular matrix
MATCH	U(1)	Compares two floating point variables
MATMLT	N	Forms the matrix product of two matrices
MATPT	U(1)	Prints lower triangular matrix
MOVEVS	U(1)	Moves observation set from variable to working storage
MTOC	N	Converts mean elements to Cartesian coordinates
MULT	U(1)	Multiplies a 3 x 3 matrix by a succession of 1 x 3 vectors
NRTPOD	U(1)	Main control for NRTPOD
NTOM	N	Reduces a row of partial derivatives to the constrained system
NXN	N	Expands the constrained solution vector and $(A^TA)^{-1}$ to the unconstrained system
OBSIN	M	Moves observations from buffer to permanent storage
OBSSRT	U(1)	Sorts observations to time sequence
OUTER	U(1)	Computes product of column and row vector
PAGE1	M*(1)	Accumulates residuals and prints
PAGE2	N	Prints the functional standard deviations
PAROUT	M*(1)	Computes residuals for residuals print
PARSET	U(1)	Initializes station data for partial derivative package
PASTOR	U(1)	Monitors residual rejection

Subroutine	Code	Functional Description
PIMOD	U(1)	Modulates an argument between 0 and 2π
PLAMDA	N	Computes partial derivatives of position, velocity, and acceleration w.r.t. current drag parameters
PLTEL	U(2)	Converts polar elements to indeterminacy free and orbital elements
POTENT	U(1)	Driver for geopotential model
POPPC	N	Computes the transformation matrix from Cartesian to orbit plane (up, down, cross) coordinates
PPLPC	M	Computes partial of ADBARV w.r.t. Cartesian
PRAUPD	M	Updates a covariance matrix to a specified time
PRAXIS	N	Computes the eigenvalues and associated engenvectors of real symmetric 3 x 3 matrix
PRCONS	U(1)	Prints program constants
PRECES	U(1)	Precesses lunar-solar ephemerides from mean of 1950.0 to true of epoch coordinates
PRELIM	M	Makes preliminary calculations in partials package
PRSSTB	M*(2)	Computes and prints mean, RMS, and number for residuals by sensor and type
PRTATA	M	Stores and prints the A^TA matrix
PRUDRV	M	Main driver for trajectory print and update package
PTOC	U(2)	Converts polar coordinates to Cartesian coordinates
PUPB	M	Computes partials of observation w.r.t. category 2 variables
RADR	M	Computes residuals; driver for partials package
RADSQ	U(1)	Computes magnitude and $(\text{magnitude})^2$ of a 3-D vector

Subroutine	Code	Functional Description
RDDATA	M	Reads NAMELIST input cards, ephemeris cards, and mean elements cards
REJECT	M*(1)	Monitors the acceptance or rejection of an observation
RMAX	M*(1)	Compares residual quantities with table of maximum values
ROTRU	U(2)	Rotates a set of vectors from mean of 1950.0 to true of date coordinates
RPRESS	U(1)	Computes perturbative acceleration due to radiation pressure
SDELET	U(1)	Moves deletion list from buffer to permanent storage
SELECT	M	Selects next observation time
SENIN	U(1)	Moves sensor data from buffer to permanent storage
SENRD	M	Reads six types of sensor cards
SETCON	M	Sets constants for program
SETIC	M	Initializes integration list
SETSTR	U(1)	Converts drag and radiation pressure parameters from external to internal units
SETTAB	M*(1)	Sets tables concerning solution vector in variable storage
SSTB	M*(2)	Accumulates sum, sum of squares, and number of residuals by sensor and data type
STEER	N	Computes the radar steering ephemeris and prints the summary values
STSMAT	U(2)	Converts upper triangular S matrix from human units to machine units
SUPMAT	U(2)	Moves input update matrix from buffer to permanent storage
TGDJD	U(2)	Converts Julian to calendar date from integration time and prints
TINIT	U(2)	Sets up initial time, computes α_{go}

Subroutine	Code	Functional Description
TIME	U(2)	Converts Y, M, D, H, M, S to Julian date: days plus fraction
TMOD	N*	Modulates an argument
TPRLM	U(2)	Sets up data for integration
TPRNT	M*(2)	Prints trajectory output
TRAJ	M	Integrates the equations of motion and variational equations of motion to a specified time
TRJGEN	M	Main driver for trajectory package
TRJGET	M	Reads trajectory record from trajectory tape from DC package
TRJOUT	N	Prepares a variable length trajectory word for the trajectory tape
TRJPRO	M	Main driver for DC, trajectory, and update interfaces
TRJTAP	M	Writes trajectory tape
UBSGET	U(1)	Gets next observation time from variable storage
UPPER	N	Converts an N x N lower triangular matrix to an upper triangular matrix with an augmented column
UVECT	U(1)	Unitizes a 3-dimensional vector
VAREQ	U(1)	Computes second derivatives of variational equations
VPERT	U(1)	Initializes variational equations
WEOFT	U(1)	Writes an ending sentinel block on observation tape
WRTOBS	U(1)	Generates observation tape
XCROSS	U(2)	Performs the cross product of two 3-dimensional vectors
YRAE	U(2)	Computes Y vector when range, azimuth, and elevation are given

5.3 BRIEF SUBROUTINE DESCRIPTIONS

Brief descriptions of subroutine changes are given in this section. In most instances, the particular modifications are extensions and/or changes in the logic rather than input/output modifications. The parenthetical code following the subroutine name refers to the reference in which the major description of the particular subroutine is given, since it does not appear in this document. There are also five library or utility type subroutines which are described here; these particular subroutines exist in the System/360 version only.

ACOS	ACOS is a double precision arc cosine routine which is present in the System/360 version of NRTPD2 only. This function (single precision) exists as a library subroutine in the 7094 version.
APPLY(1)	The logic of APPLY was extended to accommodate the extended solution vector due to the multiple ballistic coefficients and the $\ddot{R}$ sensor biases. The routine was modified for the output of mean elements in the iteration summary also.
DSQRT	DSQRT is a double precision square root routine which is present in the System/360 version of NRTPD2 only. This function (single precision) exists as a library subroutine in the 7094 version.
FIXFLT	FIXFLT is a routine which stores an integer (a fixed number) into a floating array. This subroutine is used in the System/360 version of NRTPD2 only.
FLTFIX	FLTFIX is a routine which fixes a floating number and stores it in a floating array. This subroutine is used in the System/360 version of NRTPD2 only.
IPRNT(1)	IPRNT has been modified to print the input C_DA/m table as a function of altitude on the header page.
LINES(1)	The logic of LINES has been extended to print $\ddot{R}$ residuals.
LODOBS(1)	The logic of LODOBS has been extended for the printing of $\ddot{R}$ observations.
LODSEN(1)	The logic of LODSEN has been extended for the reading, processing, and printing of the three additional sensor cards.
PAGE1(1)	PAGE1 has been modified to convert and print $\ddot{R}$ residuals.

PAROUT(1)	PAROUT has been modified for the computation of residuals.
PRSSTB(2)	PRSSTB has been modified to calculate the mean and RMS of the $\ddot{R}$ residuals by station.
REJECT(1)	The logic of REJECT has been extended to include $\ddot{R}$ residuals editing.
RMAX(1)	The logic of RMAX has been extended for the processing of $\ddot{R}$ residuals editing.
SETTAB(1)	SETTAB has been modified to include the $\ddot{R}$ bias variables.
SSTB(2)	SSTB has been modified to accumulate the sum, sum of squares, and number of residuals of $\ddot{R}$ data by station.
TMOD	TMOD is a routine which modulates argument A by argument B. This subroutine is used in the System/360 version of NRTPD2 only.
TPRNT(2)	TPRNT has been modified to print the mean elements and the steering ephemeris.

5.4 SUBROUTINE DESCRIPTIONS

SUBROUTINE IDENTIFICATION

A. Title

ASSIGN

B. Segment

NRTPOD - INPUT PROCESSOR

C. Called by subroutine

INPUT

FUNCTION

The function is to establish storage assignments for the arrays to be located in variable storage (VSTR). This routine also establishes NPR, NDPR, and NICPR.

USAGE

A. Calling sequence

Call ASSIGN

B. Input

1. COMMON

/INPP/ DATA (1000)

2. Calling sequence

—

C. Output

1. COMMON

NPR	Total number of all parameters to solve for
NDPR	Number of differential and initial parameters to solve for (Category 1)
NICPR	Number of initial condition parameters to solve for
NAROW	Starting location where one row of augmented matrix (A, B) is stored

NATA	Starting location of where the triangular A^TA is stored
NBDNS	Starting location for the bounds used by LEGS
NDPAR1 NDPAR2 NDPAR3 NDPAR4	Starting locations where the 4 sets of solution vectors will be stored
NIDLED	Starting location of where the observation deletion table begins
NIDENT	Number of entries in the NIDLED list
NIDP	Identifier for table indicating CAT 1 type variables to be solved for
NPAR	Identifies the starting location for the parameter list
NPBIS	Identifies table for current estimates of CAT 2 variables
NPRCD	Identifies table for definition of CAT 2 variables to be solved for
NR	Starting location of where the inverse A^TA is stored (in triangular form)
NRTMP	Identifies the starting location of temporary storage for special handling of the R matrix
NSCALE	Location of the list of conversion factors which convert all solution vectors and associated matrices from machine to output units and vice versa
NSTAT	Starting location of the master sensor table
VSTR	Floating point variable storage
MPR	Size of solution vector after constraint matrix has been applied. 0 if no constraints.
IMAX	Number of non-zero element of constraint matrix
CFLG	$\neq 0$ if additive constants are present in constraint problem.
MBNDS	Variable storage pointer for bounds corresponding to constrained system
NB	Variable storage pointer for constraint matrix
NC	Variable storage pointer for constraint matrix
NIJ	Variable storage pointer for indices of non-zero entries of constraint matrix
NST	Variable storage pointer for temporary cells used for linear constraints
NLAMS	Number of drag parameters in solution vector
NLID	Starting location in VSTR of the identifiers for the drag parameters appearing in the solution vector

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

NICPR	=	Number of orbital elements to solve for
NDPR	=	CAT1 variables
NPR	=	CAT1 + CAT2
NLAMS	=	Number of drag solution variables
NIDP	=	1
NLID	=	NICPR + NLAMS + NIDP
NPRCD	=	NLAMS + NLID
NIDLED	=	NPRCD + NPR - NDPR
NIJ	=	NIDLED + NPR
NPBIS	=	1
NH	=	NPR - NDPR + NPBIS
NPALM	=	2 * NLAM + NH
NPXLM	=	6 * NLAM + NPALM
NPXDLM	=	6 * NLAM + NPXLM
NAROW	=	3 * NLAM + NPXDLM
NBNDS	=	NPR + NAROW + 1
NPAR	=	NBDNS + NPR
NDPAR1	=	2 * NPR + NPAR
NDPAR2	=	NPR + NDPAR1
NDPAR3	=	NPR + NDPAR2
NDPAR4	=	NPR + NDPAR3
NSCALE	=	NPR + NDPAR4
NB	=	NSCALE + NPR
NC	=	NB + IMAX
I	=	CFLG
MBNDS	=	NC + I * NPR
NST	=	MBNDS + MPR

EQUATIONS

$$
\begin{aligned}
\mathrm{NATA} &= \mathrm{NST} + [\mathrm{MPR}\,(\mathrm{MPR} + 1)]\;/2 + \mathrm{MPR} \\
\mathrm{NR} &= [\,(\mathrm{NPR} + 1) * (\mathrm{NPR} + 2)\,]\;/2 + \mathrm{NATA} \\
\mathrm{NRTMP} &= [\,(\mathrm{NPR} + 2) * (\mathrm{NPR} + 3)\,]\;/2 - 1 + \mathrm{NR} \\
\mathrm{NSTAT} &= [\,(\mathrm{NPR} + 1) * \mathrm{NPR} \quad /2 + 1 + \mathrm{NRTMP}
\end{aligned}
$$

FLOW CHART

See EQUATIONS for order of computation.

SUBROUTINE IDENTIFICATION

A. Title

BCDOBS

B. Segment

NRTPOD - INPUT PROCESSOR

C. Called by subroutine

LODOBS

FUNCTION

To read one observation card and process the estimated standard deviations carried on the observation cards. Additional functions include processing of types 1 and 2 observation cards (Lincoln Laboratory Format) and detecting the last observation card to be processed.

USAGE

A. Calling sequence

Call BCDOBS (A, SEOF)

B. Input

1. COMMON

KOUT	Symbolic output tape number
KIN	Symbolic input tape number

2. Calling sequence

—

C. Output

1. COMMON

A(1)	Satellite ID (A)
A(2)	Year
A(3)	Month
A(4)	Day

A(5)	Hour
A(6)	Minutes
A(7)	Seconds
A(8)	Observation type
A(9)	Range - R (km) or $\ddot{R}$ (m/sec^2)
A(10)	Azimuth (deg) positive east of north
A(11)	Elevation (deg)
A(12)	Range derivative $\dot{R}$ (km/sec)
A(13)	Standard deviation of Range (km)or $\ddot{R}$ (m/sec^2)
A(14)	Standard deviation of azimuth (deg)
A(15)	Standard deviation of elevation (deg)
A(16)	Standard deviation of velocity (km/sec)

2. Calling sequence

SEOF — End of observation card read - signals end of observation data = ±1

SEOF = -1 more obs to be processed
SEOF = +1 no more obs to be processed

D. Error/action messages

1. Off line comment when program encounters type 2 observation cards:

"PROGRAM IGNORES TYPE 2 OR GREATER OBSERVATION CARDS"

2. Action

Program proceeds to process next observation card.

(A) Indicates alphanumeric

SUBROUTINE IDENTIFICATION

A. Title

BIJC

B. Segment

NRTPOD

C. Called by subroutine

INPUT

FUNCTION

BIJC sets up the constraint matrix in variable storage by defining two arrays B and IJ. If IJ(k) = 100i + j, then B(k) contains the element b_{ij}.

USAGE

A. Calling sequence

Call BIJC

B. Input

1. COMMON

VSTR, IVSTR	Variable storage
MPR	Size of constrained system
IMAX	Number of non-zero elements in constraint matrix
NIJ	Variable storage pointer for vector of coded subscripts of b_{ij} matrix
NB	Variable storage pointer for constraint matrix
NC	VSTR pointer for additive constants
CFLG	Additive constants flag

C. Output

VSTR, IVSTR	Variable storage

D. Error/action messages

—

BIJC

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

BNDSM

B. Segment

NRTPOD

C. Called by subroutine

FIT

FUNCTION

To obtain a set of bounds corresponding to an MPR x MPR system (constrained).

USAGE

A. Calling sequence

Call BNDSM

B. Input

1. COMMON

NBNDS	VSTR pointer for bounds (unconstrained)
MPR	Size of constrained system
MBNDS	VSTR pointer for bounds corresponding to constrained system
IMAX	Number of non-zero elements in constraint matrix
NIJ	IVSTR pointer for vector of coded subscripts of b_{ij}
NB	VSTR pointer for non-zero elements of constraint matrix

C. Output

1. COMMON

VSTR (MBNDS)

2. Calling sequence

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRT

B. Program

—

EQUATIONS

Given the diagonal bounds matrix β with positive elements β_k, $k = 1, \ldots, NPR$, new bounds are computed. The diagonal bounds matrix of constrained variables has elements β_j, $j = 1, \ldots, MPR$ and are computed as shown below:

$$\beta_j = \frac{1}{\sqrt{\sum_k \left(\frac{b_{kj}}{\beta_k}\right)^2}} \quad j = 1, \ldots MPR \quad MPR < NPR$$

SUBROUTINE IDENTIFICATION

A. Title

BODY

B. Segment

NRTPOD

C. Called by subroutine

DAUX

FUNCTION

The function is to compute the perturbative acceleration of a spacecraft due to other bodies in the solar system and to account for these effects in the variational equations.

USAGE

A. Calling sequence

Call BODY

B. Input

1. COMMON

TLIST	Current integration list
DBASE	Days from 1950.0 to midnight day of epoch
CMU	GM of Earth (e. r. ^{3}/min^2)
CGMR	Ratio of moon, sun GM to that of the Earth
FLVE	Flag to skip computation of variational equations
BFLAGS	Flags to indicate whether the accelerations of the moon and sun are to be considered
NDAYS	NAMELIST input variable denoting the number of days of lunar-solar ephemeris input.
NDPR	Number of CAT1 variables in solution vector

2. Calling sequence

C. Output

1. COMMON

TBPERT	The total acceleration of the vehicle due to all the desired bodies
PMAT	Matrix of the position dependent effects in the variational equations (the body effects are added to this matrix)
XN	Cartesian position of Moon and Sun

2. Calling sequence

—

SUBROUTINES USED

A. Library

—

B. Program

EVERT

RADSQ

OUTER

EQUATIONS

The position of the Moon and Sun with respect to the Earth, x_i, y_i, z_i, is obtained from the ephemeris cards.

$$R_i = \left(x_i^2 + y_i^2 + z_i^2 \right)^{1/2}$$

$$x_{vi} = x_v - x_i$$

$$y_{vi} = y_v - y_i$$

$$z_{vi} = z_v - z_i$$

where x_v, y_v, z_v is the position of the vehicle with respect to the earth.

$$R_{vi} = \left(x_{vi}^2 + y_{vi}^2 + z_{vi}^2 \right)^{1/2}$$

$$\ddot{x}_{bodies} = - \sum_{i=1}^{n} \mu_i \left[\frac{(x_v - x_i)}{R_{vi}^3} + \frac{x_i}{R_i^3} \right]$$

$$\ddot{y}_{bodies} = - \sum_{i=1}^{u} \mu_i \left[\frac{(y_v - y_i)}{R_{vi}^3} + \frac{y_i}{R_i^3} \right]$$

$$\ddot{z}_{bodies} = - \sum_{i=1}^{u} \mu_i \left[\frac{(z_v - z_i)}{R_{vi}^3} + \frac{z_i}{R_i^3} \right]$$

$$\text{PMAT} = \text{PMAT} + \begin{bmatrix} \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}^2}{R_{vi}^5} - \frac{1}{R_{vi}^3} \right) & \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, y_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, z_{vi}}{R_{vi}^5} \right) \\ \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, y_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{u} \mu_i \left(\frac{3y_{vi}^2}{R_{vi}^5} - \frac{1}{R_{vi}^3} \right) & \sum_{i=1}^{u} \mu_i \left(\frac{3y_{vi}\, z_{vi}}{R_{vi}^5} \right) \\ \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, z_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{u} \mu_i \left(\frac{3y_{vi}\, z_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{u} \mu_i \left(\frac{3z_{vi}^2}{R_{vi}^5} - \frac{1}{R_{vi}^3} \right) \end{bmatrix}$$

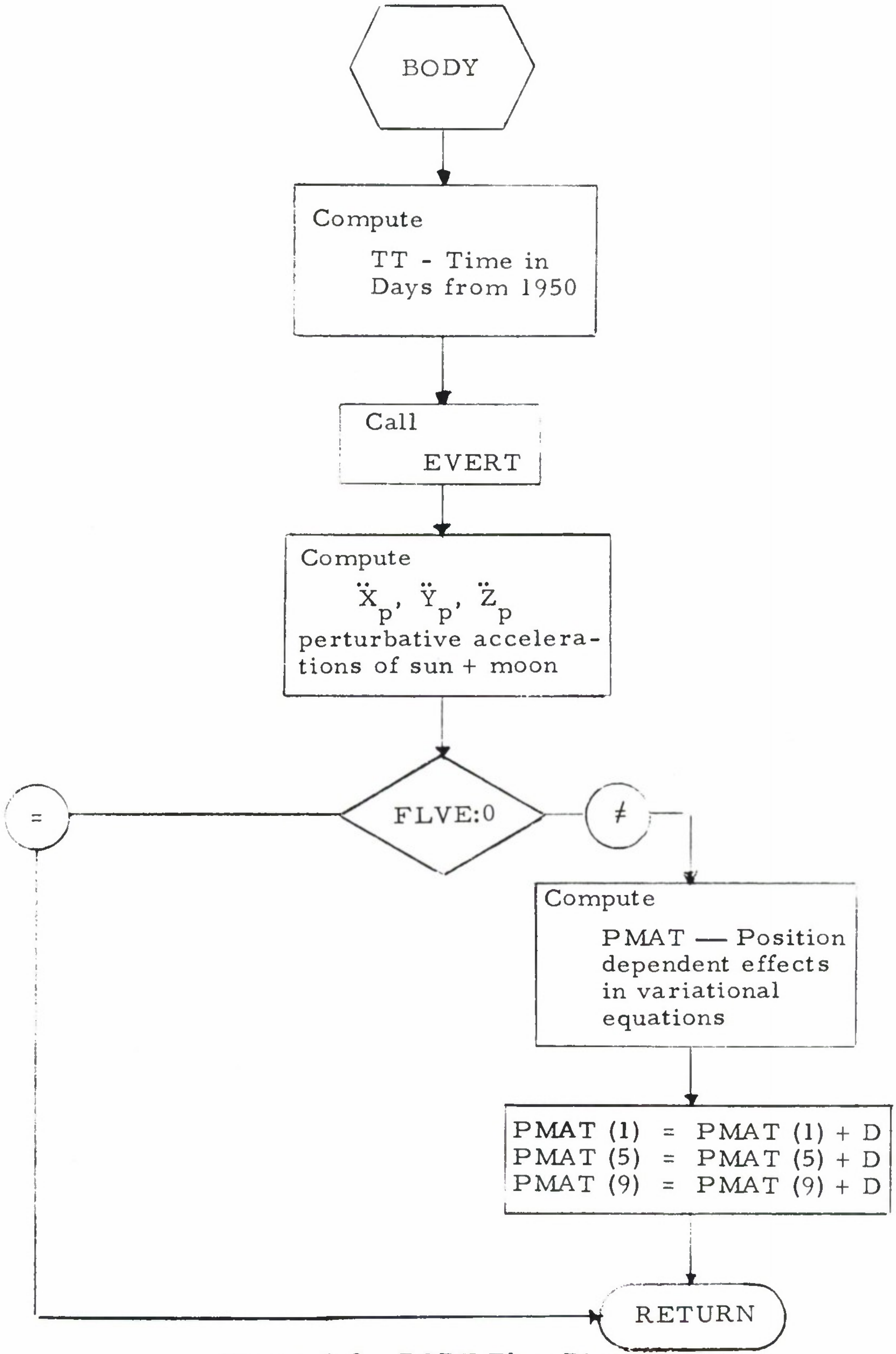


Figure 5-2. BODY Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

CONST

B. Segment

NRTPOD

C. Called by subroutine

INPUT

FUNCTION

Subroutine CONST computes those constants which are functions of other constants that may be altered at input time.

USAGE

A. Calling sequence

CALL CONST

B. Input

1. COMMON

CKMER	Conversion from earth radii to kilometers
CFTER	Conversion from earth radii to feet
CFTNM	Conversion from nautical miles to feet
CELLIP	Ellipticity of the earth

2. Calling sequence

C. Output

1. COMMON

CMTER	Conversion from earth radii to meters
CNMER	Conversion from earth radii to nautical miles
CBE	Semi-minor axis of the earth (earth radii)
TRM1	First term used in the computation of the "radius at sea level" (See equations.)
TRM2	Second term used in the computation of the "radius at sea level" (See equations.)

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library
EXP

B. Program
—

EQUATIONS

CMTER = CKMER * 1000.

CNMER = CFTER/CFTNM

CBE = 1.0 - CELLIP

The distance from the center of the earth to the surface of the ellipsoid at geocentric latitude φ, termed "radius at sea level", is given by,

$$\rho = a_e (1 + k \sin^2 \varphi)^{-1/2}$$

where a_e is the earth equatorial radius and

$$k = \frac{2 - \epsilon}{(1 - \epsilon)^2} \epsilon$$

is a constant derived from ellipticity, ϵ.

Utilizing a series expansion along with an introduction of a shifted Chebyshev polynomial of order three, an approximation for the radius at sea level is obtained.

$$\rho \approx a_e \left[1 - \left(\frac{k}{2} - \frac{45}{256} k^3 \right) S^2 + \left(\frac{3}{8} k^2 - \frac{15}{32} k^3 \right) S^4 \right]$$

where

$$\sin \varphi = S$$

This routine computes the constant terms in the approximation for ρ

$$TRM1 = \frac{k}{2} - \frac{45}{256} k^3$$

$$TRM2 = \frac{3}{8} k^2 - \frac{15}{32} k^3$$

SUBROUTINE IDENTIFICATION

A. Title

CTOM

B. Segment

NRTPOD

C. Called by subroutines

APPLY
TPRNT

FUNCTION

To convert a set of Cartesian elements to osculating elements and then to mean elements.

USAGE

A. Calling sequence

Call CTOM(TNOMX, ADBAR, ITER)

B. Input

1. COMMON

CJ2	J2 Earth Harmonic
CMU	μ (Earth Radii, Minutes)
CPI	π
C2PI	2π
KOUT	Output tape number

2. Calling sequence

TNOMX(1)	x	Earth Radii
TNOMX(2)	y	Earth Radii
TNOMX(3)	z	Earth Radii
TNOMX(4)	$\dot{x}$	Earth Radii per Minute
TNOMX(5)	$\dot{y}$	Earth Radii per Minute
TNOMX(6)	$\dot{z}$	Earth Radii per Minute
ITER	Number of iterations to be used to calculate δ's; see subroutine DLSTV	

C. Output

1. COMMON

None

2. Calling sequence

ADBAR(1)	a_{K-25_m}	(Earth Radii)
ADBAR(2)	e_m	
ADBAR(3)	i_m	(Radians)
ADBAR(4)	Ω_m	(Radians)
ADBAR(5)	ω_m	(Radians)
ADBAR(6)	M_m	(Radians)
ADBAR(7)	$\dot{\omega}_m$	(Radians / Minute)
ADBAR(8)	$\dot{\Omega}_m$	(Radians / Minute)

D. Error Messages

If E (eccentric anomaly) fails to converge after 50 iterations

E FAILED TO CONVERGE

THE VALUE OF E IS ±. XXXXXE ±XX

The computation proceeds with the last computed value of E.

SUBROUTINES USED

A. Library

ABS
SQRT
ATNQ
SIN
COS

B. Program

PIMOD	To set the principle value of an angle between 0 and 2π
DLSTV	To compute the δ's for conversion from osculating to mean and mean to osculating

EQUATIONS

1. Compute epoch values of

a) magnitude of radius vector

$$r_o = \sqrt{x^2 + y^2 + z^2}$$

b) Angular momentum

$$h^2_o = (y\dot{z} - z\dot{y})^2 + (z\dot{x} - x\dot{z})^2 + (x\dot{y} - y\dot{x})^2$$

c) Orbital semi-parameter

$$p_o = h_o^2/\mu$$

2. Compute osculating orbital elements.

a) Semi-major axis

$$a_{os} = r_o\mu/\left[2\mu - r_o(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)\right]$$

b) Eccentricity

$$e_{os} = +\sqrt{(p_o/r_o - 1)^2 + \frac{p_o}{\mu r_o^2}(x\dot{x} + y\dot{y} + z\dot{z})^2}$$

c) The true anomaly v_o

$$e_{os}\cos v_o = (p_o/r_o - 1)$$

$$e_{os}\sin v_o = \sqrt{\frac{p_o}{\mu}}\left(\frac{x\dot{x} + y\dot{y} + z\dot{z}}{r_o}\right)$$

d) The orbital inclination and the longitude of the ascending node

$$\sin i_{os}\sin\Omega_{os} = \frac{y\dot{z} - z\dot{y}}{h_o}$$

$$\sin i_{os}\cos\Omega_{os} = \frac{x\dot{z} - z\dot{x}}{h_o}$$

$$\cos i_{os} = \frac{x\dot{y} - y\dot{x}}{h_o}$$

e) The argument of latitude, u, is determined from

$$\cos u_o = \frac{x}{r_o}\cos\Omega_{os} + \frac{y}{r_o}\sin\Omega_{os}$$

$$\sin u_o = z/(r_o \sin i_{os})$$

f) The argument of perigee

$$\omega_{os} = u_o - v_o$$

g) The eccentric anomaly

$$e_{os} \sin E_o = (x\dot{x} + y\dot{y} + z\dot{z}) / \sqrt{\mu a_{os}}$$

$$e_{os} \cos E_o = (1 - r_o / a_{os})$$

h) The mean anomaly

$$M_{os} = E_o - e_{os} \sin E_o$$

Compute the initial K-25 element where $A_2 = \frac{3}{2} J_2 a_e^2$

$$a_{os_{K-25}} = a_{os} \left[1 - \frac{A_2}{p_o^2} \left(1 - \frac{3}{2} \sin^2 i_{os} \right) \left(\sqrt{1 - e_{os}^2} \right) \right]$$

Compute the initial K-25 p

$$p_o = a_{os_{K-25}} \left(1 - e^2_{os} \right)$$

Then iterate on the following index, k

$$a_{K-25_k} = a_k \left[1 - \frac{A_2}{p_{k-1}^2} \left(1 - \frac{3}{2} \sin^2 i_k \right) \left(\sqrt{1 - e_k^2} \right) \right]$$

$$p_k = a_{K-25_k} \left(1 - e^2_k \right)$$

Compute δ's using $(a_{K-25_k}, e_k, i_k, \Omega_k, \omega_k, M_k)$

Compute
$$a_k = a_{os} - \delta_{a_k}$$

$$e_k = e_{os} - \delta_{e_k}$$

$$i_k = i_{os} - \delta_{i_k}$$

$$\Omega_k = \Omega_{os} - \delta_{\Omega_k}$$

$$\omega_k = \omega_{os} - \delta_{\omega_k}$$

$$M_k = M_{os} - \delta_{M_k}$$

$r = r_o$, Iterate Kepler's equations (see MTOC) to find E and v after each iteration. After the last iteration, the mean values are:

$$a_{K-25_m} = a_{K-25_k}$$

$$e_m = e_k$$

$$i_m = i_k$$

$$\Omega_m = \Omega_k$$

$$\omega_m = \omega_k$$

$$M_m = M_k$$

After iterating compute the secular rates of ω and Ω

$$\dot{\omega}_m = \frac{A_2}{p_k^2} \sqrt{\frac{\mu}{a_{K-25_k}^3}} \left(2 - \frac{5}{2} \sin^2 i_k\right)$$

$$\left\{1 - \frac{A_2}{p_k^2} \left(1 - \frac{3}{2} \sin^2 i_k\right) \sqrt{1 - e_k^2}\right\}^{1/2}$$

$$\dot{\Omega}_m = - \frac{A_2}{p_k^2} \sqrt{\frac{\mu}{a_{K-25_k}^3}} \cos i_k$$

$$\left\{1 - \frac{A_2}{p_k^2} \left(1 - \frac{3}{2} \sin^2 i_k\right) \sqrt{1 - e_k^2}\right\}^{1/2}$$

SUBROUTINE IDENTIFICATION

A. Title

DAUX

B. Segment

NRTPOD

C. Called by subroutine

TRAJ
SETIC

FUNCTION

The function is to compute the second derivatives in the equations of motion and control the computation of the second derivatives in the variational equations.

USAGE

A. Calling sequence

Call DAUX

B. Input

1. COMMON

TLIST	Numerical integration working storage
SGAMAM	Constant used in calculating radiation pressure effects, $S\gamma A/m$ (e.r.^{3}/min^2)
CDAD2M	Drag parameter $C_DA/2m$ (ft^2/slug)
FLVE	Variational equation control flag $\neq 0$ compute variational equations
TBPERT	Acceleration due to bodies (e.r./min^2)
TPOT	Acceleration due to aspherical potential (er/min^2)
TDRAG	Acceleration due to drag (e.r./min^2)
TRPRES	Acceleration due to radiation pressure (e.r./min^2)
TR	Magnitude of geocentric position vector, R(e. r.)
CMU	GM earth (e.r.^{3}/min^2)
NDPR	Total number of CATEGORY 1 variables to solve for
NLAM	Total number of entries in the altitude C_DA/m table

TR2	R^2
TR3	R^3
TR5	R^5
TR7	R^7

2. Calling sequence

—

C. Output

1. COMMON

TLIST (58-60) Numerical integration working storage containing the total acceleration

2. Calling sequence

—

SUBROUTINES USED

A. Library

—

B. Program

BODY

DRAG

POTENT

RADSQ

RPRESS

VAREQ

EQUATIONS

The Cowell formulation of the equations of motion is used:

$$R = \left(x^2 + y^2 + z^2\right)^{1/2}$$

$$\ddot{x} = \frac{-\mu\ x}{R^3} + \ddot{x}_{\text{bodies}} + \ddot{x}_{\text{drag}} + \ddot{x}_{\text{potential}} + \ddot{x}_{\text{radiation pressure}}$$

$$\ddot{y} = \frac{-\mu\ y}{R^3} + \ddot{y}_{\text{bodies}} + \ddot{y}_{\text{drag}} + \ddot{y}_{\text{potential}} + \ddot{y}_{\text{radiation pressure}}$$

$$\ddot{z} = \frac{-\mu\ z}{R^3} + \ddot{z}_{\text{bodies}} + \ddot{z}_{\text{drag}} + \ddot{z}_{\text{potential}} + \ddot{z}_{\text{radiation pressure}}$$

DAUX

where

$\ddot{x}_{bodies}$ = The perturbation acceleration due to other bodies in the solar system

$\ddot{x}_{drag}$ = The perturbation acceleration due to atmosphere drag

$\ddot{x}_{potential}$ = The perturbation acceleration due to the potential field set by the aspherical earth

$\ddot{x}_{radiation\ pressure}$ = The perturbation acceleration due to solar radiation pressure

The tests are made to see which of the above perturbation effects are to be included in the evaluation of the equations of motion.

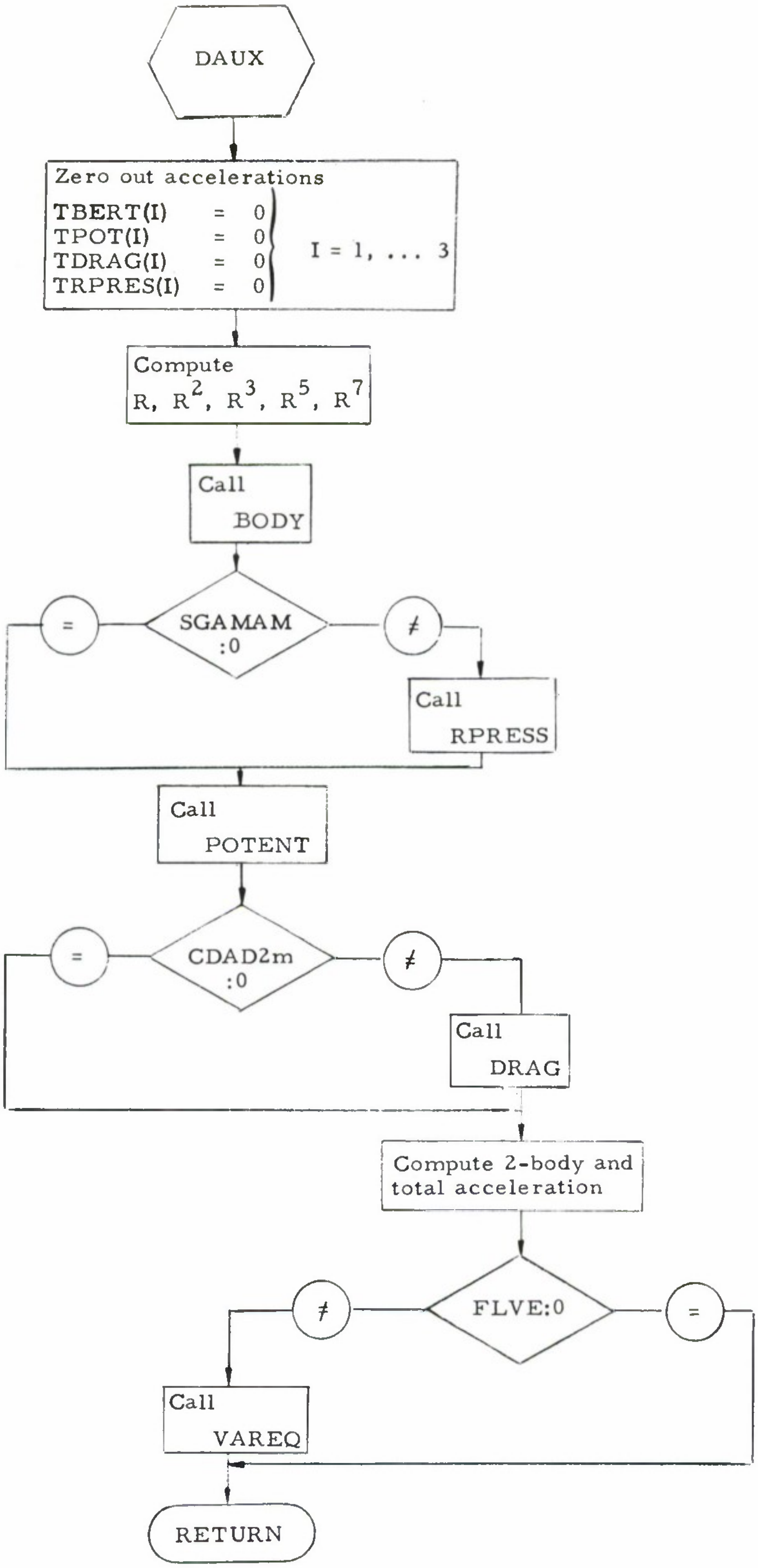


Figure 5-3. DAUX Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

DLSTV

B. Segment

NRTPOD

C. Called by subroutines

CTOM
MTOC

FUNCTION

To compute the differentials used to convert from osculating to mean and mean to osculating.

USAGE

A. Calling sequence

Call DLSTV (STATE, R, V, E, DELTA)

B. Input

1. COMMON

CJ2 J2 Earth Harmonic

2. Calling sequence

	Osculating or	Mean	
STATE(1)	$a_{o_{K-25}}$	$a_{m_{K-25}}$	(Earth Radii)
STATE(2)	e_o	e_m	
STATE(3)	i_o	i_m	(Radians)
STATE(4)	Ω_o	Ω_m	(Radians)
STATE(5)	ω_o	ω_m	(Radians)
STATE(6)	M_o	M_m	(Radians)
R	Magnitude of radius vector		(Earth Radii)
V	True anomaly		(Radians)
E	Eccentric anomaly		(Radians)

C. Output

1. COMMON

None

2. Calling sequence

DELTA(1) δ_a

DELTA(2) δ_e

DELTA(3) δ_i

DELTA(4) δ_Ω

DELTA(5) δ_ω

DELTA(6) δ_M

D. Error Messages

None

SUBROUTINES USED

A. Library

SIN
COS
SQRT

B. Program

None

EQUATIONS

Equations (2), (5), and (6) have been formulated to preserve numerical accuracy when eccentricity is near zero, and hence do not appear as in the standard references, Kozai, et al.

$$da = \frac{A_2}{a} \left[\frac{2}{3} \left(1 - \frac{3}{2} \sin^2 i\right) \left\{ \left(\frac{a}{r}\right)^3 - \left(1 - e^2\right)^{-3/2} \right\} + \left(\frac{a}{r}\right)^3 \sin^2 i \cos 2(v + \omega) \right] \tag{1}$$

$$de = \frac{A_2}{a^2} \left\{ (1 - e^2) \left[\frac{1}{3} \left(1 - \frac{3}{2} \sin^2 i\right) (S_5 - S_3) \right] + \frac{\sin^2 i \cos 2u}{2} S_6 - \frac{\sin^2 i}{2(1-e^2)} \left[\cos (v + 2\omega) + \frac{1}{3} \cos (3v + 2\omega) \right] \right\} \tag{2}$$

where

$$S_1 = e/\left[1 + (1 - e^2)^{1/2}\right]$$

$$S_2 = S_1/(1 - S_1 e)$$

$$S_3 = 3S_2 + 3S_2^{\,2} e + S_2^{\,3} e^2$$

$$S_4 = \cos E/(1 - e \cos E)$$

$$S_5 = 3S_4 + 3S_4^{\,2} e + S_4^{\,3} e^2$$

$$S_6 = \frac{S_5 + (1 + S_5 e)(e^3 - 2e)}{(1 - e^2)}$$

$$di = \frac{1}{4}\frac{A_2}{p^2} \sin 2i \left\{\cos 2(v + \omega) + e \cos (v + 2\omega) + \frac{e}{3} \cos (3v + 2\omega)\right\} \tag{3}$$

where

$$p = a(1 - e^2)$$

$$d\Omega = -\frac{A_2}{p^2} \cos i \left\{(v - M) - \frac{1}{2} \sin 2(v + \omega) + e \sin v - \frac{e}{2} \sin (v + 2\omega) - \frac{e}{6} \sin (3v + 2\omega)\right\} \tag{4}$$

$$d\omega = \frac{A_2}{p^2}\left[A_\omega(e^{-1}) + B_\omega(e^0) + C_\omega(e^1)\right] \tag{5}$$

where

$$A_\omega(e^{-1}) = \frac{1}{12e}\left[12 \sin v + \sin^2 i \left\{7 \sin (3v + 2\omega) - 3 \sin (v + 2\omega) - 18 \sin v\right\}\right]$$

$$B_\omega(e^o) = \frac{1}{8}\Big[4 \sin 2v + 16(v - M) - 4 \sin 2(v + \omega)$$

$$+ \sin^2 i \Big\{3 \sin(4 v + 2\omega) - 6 \sin 2 v$$

$$+ 10 \sin 2(v + \omega) - 20(v - M)\Big\}\Big]$$

$$C_\omega(e^1) = \frac{e}{48}\Big[4 \sin 3v + 84 \sin v - 24 \sin (v + \omega)$$

$$- 8 \sin (3v + 2\omega) + \sin^2 i \Big\{3 \sin (5v + 2\omega)$$

$$+ 3 \sin (v - 2\omega) - 6 \sin 3v$$

$$+ 19 \sin (3v + 2\omega) + 45 \sin (v + 2\omega)$$

$$- 102 \sin v\Big\}\Big]$$

$$dM = \frac{A_2}{p^2} \sqrt{1 - e^2} \Big[A_M(e^{-1}) + B_M(e^o) + C_M(e^1)\Big] \tag{6}$$

where

$$A_M(e^{-1}) = - \frac{1}{12e}\Big[12 \sin v + \sin^2 i \Big\{7 \sin (3v + 2\omega)$$

$$- 3 \sin (v + 2\omega) - 18 \sin v\Big\}\Big]$$

$$B_M(e^o) = - \frac{1}{8}\Big[4 \sin 2v + \sin^2 i \Big\{3 \sin (4v + 2\omega)$$

$$- 6 \sin 2v\Big\}\Big]$$

$$C_M(e^1) = - \frac{e}{48}\Big[4 \sin 3v - 12 \sin v$$

$$+ \sin^2 i \Big\{3 \sin (5v + 2\omega) + 3 \sin (v - 2\omega)$$

$$- 6 \sin 3v - \sin (3v + 2\omega)$$

$$- 15 \sin (v + 2\omega) + 18 \sin v\Big\}\Big]$$

SUBROUTINE IDENTIFICATION

A. Title

DPRLM

B. Segment

NRTPOD - Input Processor

C. Called by subroutine

INPUT

FUNCTION

To set up preliminary information for the input processor link. This information concerns epoch time and mode of epoch position and velocity.

USAGE

A. Calling sequence

Call DPRLM

B. Input

1. COMMON

CDEG	Degrees/radian
CWE	Earth's rotational rate (radians/min)
STVEC	Input initial conditions
DTYPE	Initial conditions type
DAYINT	Integer portion of Julian date
DAYFRC	Fractional portion of Julian date
TNULL	Time to which input elements are to be updated
SMELM	21-word vector containing the Smithsonian mean elements and their time derivatives; see Table I in MTOC Subroutine.
CWE	Earth's rotational rate
TEPOCH	Time of epoch, minutes from 0 hours

TALFAG	Right ascension of Greenwich at 0 hour day of epoch
STVEC	Input initial conditions - Cartesian or polar

2. Calling sequence

—

C. Output

1. COMMON

TALFAG	α_g for midnight day of epoch
TEPOCH	Epoch time, minutes from midnight
TNOMX	Initial Cartesian coordinates
TNOMP	Initial spherical coordinates

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

TINIT	Sets up initial time, computes α_g and DBASE (days from 1950 to day of epoch)
PIMOD	Takes principle value of angle between 0 and 2π
PTOC	Converts from polar coordinates to Cartesian coordinates
CTOP	Converts Cartesian coordinates to polar coordinates
SETSTR	Sets up drag, radiation pressure, potential, parameters
IPRNT	Prints header page
MTOC	Converts mean elements to Cartesian coordinates

SUBROUTINE IDENTIFICATION

A. Title

DPROS

B. Segment

NRTPOD

C. Called by subroutine

INPUT

FUNCTION

To issue calls on the sensor and observation loading routines if required by input.

USAGE

A. Calling sequence

CALL DPROS

B. Input

1. COMMON

PREFLG	NRTPOD control flags (JDC columns 31-40)
MT	Logical unit for the observation tape
NLAM	Total number of entries in the C_DA/m table
ALTS	Altitude table for multiple drag (kilometers)
CLAMDA	C_DA/m table corresponding to ALTS table (meters ^{2}/kilogram)

2. Calling sequence

—

C. Output

1. COMMON

—

2. Calling sequence

—

D. Error/action messages

—

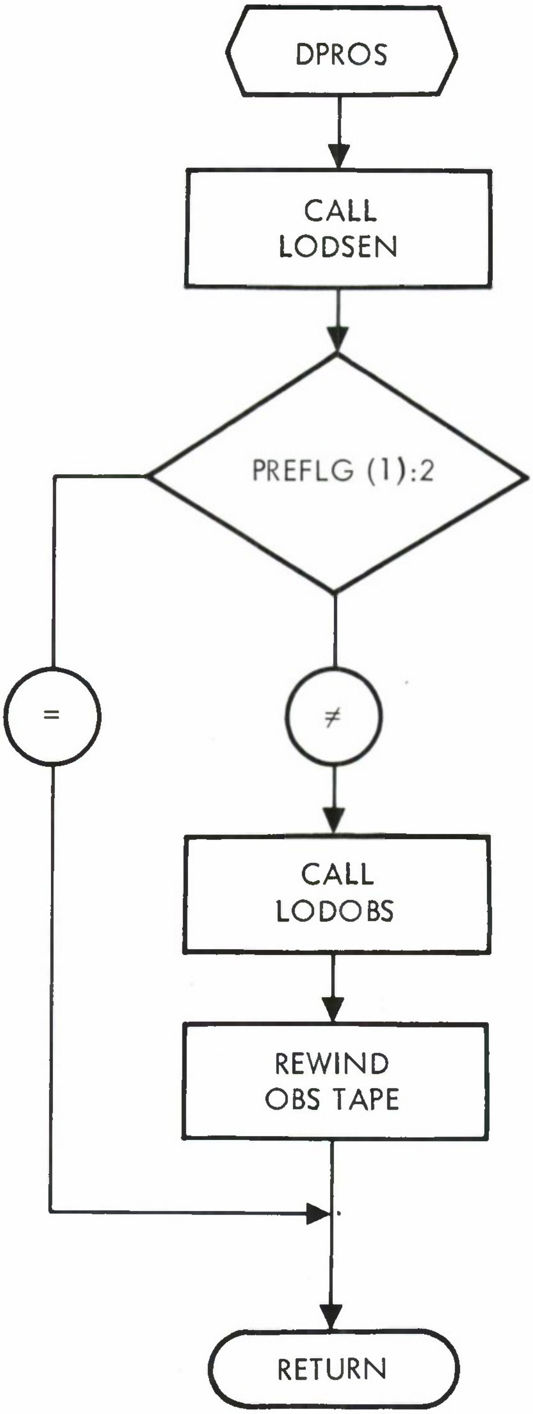


Figure 5-4. DPROS Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

DRAG

B. Segment

NRTPOD

C. Called by subroutine

DAUX

FUNCTION

Function is to compute the perturbative acceleration of a vehicle due to atmosphere drag and to account for these effects in the variational equations.

USAGE

A. Calling sequence

Call DRAG

B. Input

1. COMMON

FLVE	Variational equation control flag
TV	Earth-fixed velocity of vehicle
TVA	Magnitude of TV
CELLIP	Constant = ellipticity of the Earth
TLIST	Numerical integration working storage
TR2	Square of TR
TR	Magnitude of the vector from the center of the Earth to the vehicle
CWE	Constant = rotation rate of the earth (radians/minutes) = ω_e
CDAD2M	Drag parameters $C_D A/2m$
CFTER	Feet per earth radius
TRHOA	Density in slugs/ft^3
TALT	Altitude of vehicle in feet

2. Calling sequence

—

C. Output

1. COMMON

TDRAG	Perturbative acceleration due to drag
VMAT	Matrix of velocity dependent terms in the evaluation of the variational equations
PMAT	Matrix of position dependent terms in the evaluation of the variational equation. (The drag effects are added to the contents of this matrix.)

D. Error/ action messages

--

SUBROUTINES USED

A. Library

SQRT

B. Program

JACHIA

OUTER

EQUATIONS

$$R_e = \frac{1 - \epsilon}{\left[1 - \epsilon(2 - \epsilon)\left(\frac{x^2 + y^2}{R^2}\right)\right]^{1/2}} = \text{radius of the Earth}$$

$$\text{Altitude} = R - R_e$$

$$\rho_a = \text{density at the given altitude}$$

$$\left.\begin{aligned} v_{ax} &= \dot{x} + \omega_e y \\ v_{ay} &= \dot{y} - \omega_e x \\ v_{az} &= \dot{z} \end{aligned}\right\} \quad \text{Earth-fixed velocity}$$

$$v_a = \left(v_{ax}^2 + v_{ax}^2 + v_{ax}^2\right)^{1/2}$$

$$\lambda = \frac{C_d A}{2m}$$

$$\ddot{x}_{drag} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{ax}$$

$$\ddot{y}_{drag} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{ay}$$

$$\ddot{z}_{drag} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{az}$$

$$PMAT = PMAT - \lambda\rho_a v_a \begin{bmatrix} 0 & \omega_e & 0 \\ -\omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{\lambda V_a \rho'}{R} \begin{bmatrix} v_{ax}x & v_{ax}y & v_{ax}z \\ v_{ay}x & v_{ay}y & v_{ay}z \\ v_{az}x & v_{az}y & v_{az}z \end{bmatrix}$$

$$- \frac{\lambda\rho_a}{V_a} \begin{bmatrix} v_{ax}^2 & v_{ax}v_{ay} & v_{ax}v_{az} \\ v_{ax}v_{ay} & v_{ay}^2 & v_{ay}v_{az} \\ v_{ax}v_{az} & v_{ay}v_{az} & v_{az}^2 \end{bmatrix} \begin{bmatrix} 0 & \omega_e & 0 \\ -\omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{-\partial\lambda}{\partial h} \quad \frac{\rho_a v_a}{r} \begin{bmatrix} v_{ax}x & v_{ax}y & v_{ax}z \\ v_{ay}x & v_{ay}y & v_{ay}z \\ v_{az}x & v_{az}y & v_{az}z \end{bmatrix}$$

$$VMAT = VMAT - \frac{\lambda\rho_a}{V_a} \begin{bmatrix} v_{ax}^2 & v_{ax}v_{ay} & v_{ax}v_{az} \\ v_{ax}v_{ay} & v_{ay}^2 & v_{ay}v_{az} \\ v_{ax}v_{az} & v_{ay}v_{az} & v_{az}^2 \end{bmatrix} - \lambda\rho_a v_a \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

When multiple $C_D A/m$'s are included in the solution vector, DRAG computes the inhomogeneous terms in the variational equations of drag. These equations are derived in Reference 3. The inhomogeneous term is expressed as:

$$\frac{\partial \lambda}{\partial \lambda_i} \rho \bar{v}_a v_a$$

Written out in full, this term has the form

$$\frac{\partial \lambda}{\partial \lambda_i} \rho \bar{v}_a v_a = E_i(h) \rho \bar{v}_a v_a$$

where

$$E_i(h) = \begin{cases} 0 & h < h_{i-1} \\ \dfrac{h - h_{i-1}}{h_i - h_{i-1}} & h_{i-1} < h < h_i \\ \dfrac{h_{i+1} - h}{h_{i+1} - h_i} & h_i < h < h_{i+1} \\ 0 & h_{i+1} < h \end{cases}$$

SUBROUTINE IDENTIFICATION

A. Title
DRDP

B. Segment
NRTPOD

C. Called by subroutine
RADR

FUNCTION

Function is to compute the partial of the M^{th} type of observation with respect to the solution vector.

USAGE

A. Calling sequence
Call DRDP (M)

B. Input

1. COMMON

NAROW	Starting location where one row of the augmented matrix (A, B) is stored
NDPR	Number of all differential plus initial parameters to solve for (Category 1)
PUBS	Current observation buffer; ID, time, R or $\ddot{R}$, A, E, $\dot{R}$, type
PCSA	Cos A
PRSUB1	R_1 = VR
PSNA	Sin A
PSNE	Sin E
PSTAT	Working storage for sensor information
PUDTI	$\vec{\dot{u}} = (\dot{u}_1, \dot{u}_2, \dot{u}_3)$
PUI	(u_1, u_2, u_3)
PWDTPP	$\partial \dot{w} / \partial P_i$

PWPP $\partial w/\partial P_i$

PWDT2P $\partial \ddot{w}/\partial P_i$

PUDDTI $\vec{\ddot{u}} = (\ddot{u}_1, \ddot{u}_2, \ddot{u}_3)$

2. Calling sequence

M Observation type number (1, 2, 3, 4,)

1 R or $\ddot{R}$ (range or range acceleration)

2 Azimuth

3 Elevation

4 $\dot{R}$ (range rate)

C. Output

1. COMMON

VSTR(NAROW) ➤ VSTR(NAROW + NDPR - 1) contains the partial derivatives of the Mth type observation with respect to the Category 1 variables being solved for

2. Calling sequence

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

Range (type 0 observation, M = 1)

$$\frac{\partial R}{\partial p_i} = u_1 \frac{\partial w_1}{\partial p_i} + u_2 \frac{\partial w_2}{\partial p_i} + u_3 \frac{\partial w_3}{\partial p_i} \quad p_i \rightarrow \alpha, \delta, \beta, A, r, v, \lambda_1, \lambda_2 \ldots \lambda_n$$

Azimuth (type 0 observation, M = 2)

$$\frac{\partial A}{\partial p_i} = \frac{1}{R_1} \left[\frac{\partial w_2}{\partial p_i} \cos A - \left(\frac{\partial w_1}{\partial p_i} \sin \phi^* + \frac{\partial w_3}{\partial p_i} \cos \phi^* \right) \sin A \right]$$

Elevation (type 0 observation, M = 3)

$$\frac{\partial E}{\partial p_i} = \frac{1}{R_1}\left(\frac{\partial w_1}{\partial p_i}\cos\phi^* + \frac{\partial w_3}{\partial p_i}\sin\phi^* - \frac{\partial R}{\partial p_i}\sin E\right)$$

Range Rate (type 0 observation, M = 4)

$$\frac{\partial \dot{R}}{\partial p_i} = \left(\frac{\partial \overline{w}}{\partial p_i} \cdot \dot{\overline{u}}\right) + \left(\overline{u} \cdot \frac{\partial \dot{\overline{w}}}{\partial p_i}\right)$$

Range Acceleration (type 1 observation, M = 1)

$$\frac{\partial \ddot{R}}{\partial p_i} = \frac{\partial}{\partial t}\left(\frac{\partial \dot{R}}{\partial p_i}\right) = \ddot{\vec{u}} \cdot \frac{\partial \overline{w}}{\partial p_i} + 2\,\overline{\dot{u}} \cdot \frac{\partial \overline{\dot{w}}}{\partial p_i} + \overline{u} \cdot \frac{\partial \overline{\ddot{w}}}{\partial p_i}$$

SUBROUTINE IDENTIFICATION

A. Title
ELEM

B. Segment
NRTPOD

C. Called by subroutine
NXN

FUNCTION

To access an element of a lower triangular matrix stored by rows.

USAGE

A. Calling sequence

S_{ij} = RLRM(S, I, J)

B. Input

1. Calling sequence

S - Location of s

I - Row number

J - Column number

C. Output

1. COMMON

ELEM - Element S_{ij}

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

EQUATIONS

$$k = i(i - 1)/2 + j$$

$$ELEM = S(k)$$

SUBROUTINE IDENTIFICATION

A. Title

FALSI

B. Segment

NRTPOD

C. Called by subroutine

TRJGEN

FUNCTION

Subroutine FALSI utilizes the method of "false position" (regula falsi), in determining altitude cutoffs as a function of time.

USAGE

A. Calling sequence

Call FALSI

B. Input

1. COMMON

TG	Integration time to go ... minutes from 0 hrs day of epoch
TLIST	Integration list (See TRAJ subroutine.)
TRAJX	Integration coordinates - referenced to time, TG
TCRASH	Impact flag - non-zero if earth impact has occurred
TEPOCH	Time of epoch, minutes from 0 hrs. day of epoch
CHEPS	Tolerance criterion of altitude cutoffs (earth radii)
TMINUS	Flag indicating the direction of integration to subroutine SELECT

2. Calling sequence

—

C. Output

1. COMMON

ALT	Two altitude layers bounding the current region of influence of the drag coefficients (C_DA/m)
INFG	Flag indicating to subroutine PLAMDA whether an altitude crossing has occurred and which region of drag influence has been entered INFG = 0 no altitude crossing occurred = 1 vehicle has reentered altitude region 1 = 2 vehicle has left influence of altitude region 1 and crossed into altitude region 2
IFVE(2)	Of the two inhomogeneous variational systems inside any one altitude division being integrated, IFVE cells flag which of the drag coefficients in the current region of influence are being solved for IFVE(I) = 0 Ith inhomogeneous variational system is not being solved for IFVE(I) ≠ 0 Ith inhomogeneous variational system is being solved for where I = 1, 2

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

SIGN	Sign of arg 2 times |arg 1|
ABS	Absolute value

B. Program

DAUX	Driver controlling routines which compute the accelerations in the equations of motion and accelerations in the variational equations
FVE	Determines the flags which indicate the C_DA/m's to be solved in the current altitude region of influence
HEIGHT	Computes altitude of vehicle above the reference ellipsoid
HINT	Computes the current two altitude layers that the vehicle lies between during the trajectory simulation
PLAMDA	Computes the partials of x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$, $\ddot{x}$, $\ddot{y}$, $\ddot{z}$, with respect to each C_DA/m to be solved for
TRAJ	Integrates the equations of motion and up to 24 variational equations to a specified time

EQUATIONS

In the method of "false position" (regula falsi), the iteration is initiated by finding z_o and z_1 such that f_o and f_1 are of opposite signs and by defining the slope of the secant P_oP_1, so that

$$z_2 = z_1 - \left(\frac{z_1 - z_o}{f_1 - f_o}\right) f_1 = \frac{f_1 z_o - f_o z_1}{f_1 - f_o}$$

In each following iteration, the slope is taken as the slope of the line joining P_k and the most recently determined point at which the ordinate differs in sign from that at P_k. See figure 5-5.

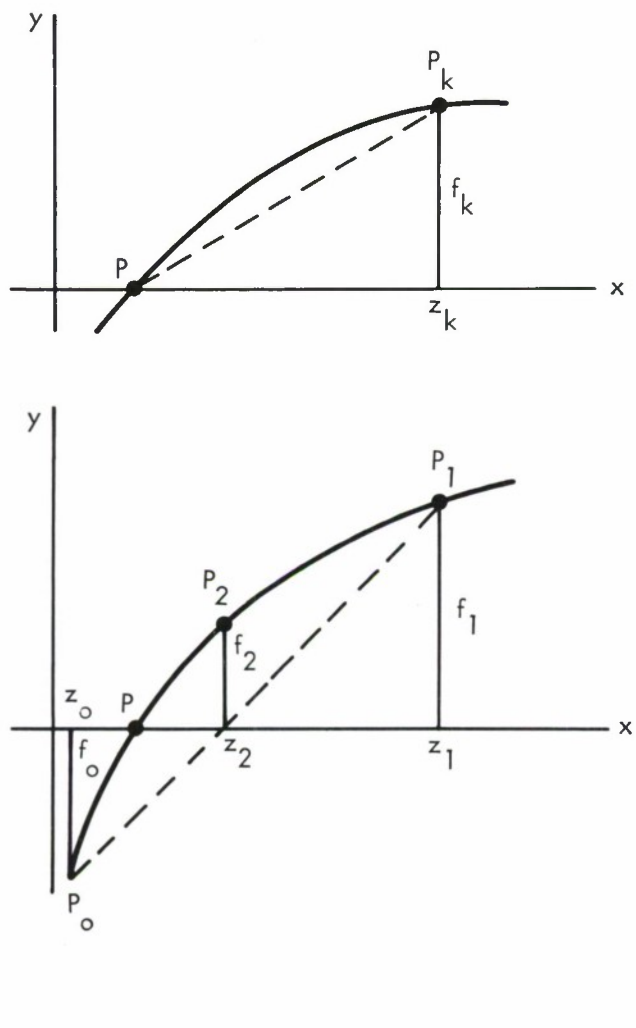


Figure 5-5. Iteration Techniques Used in Subroutine FALSI

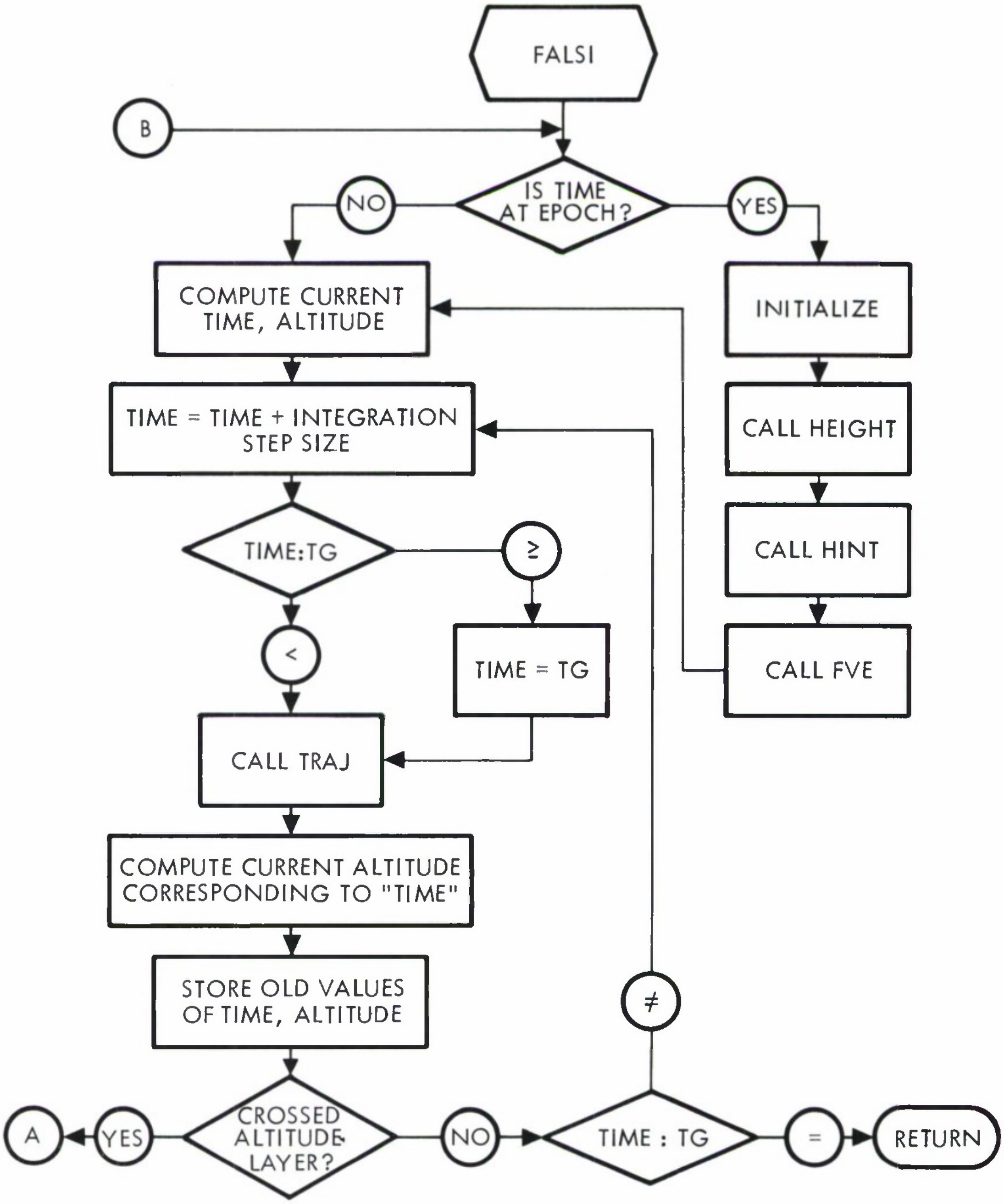


Figure 5-6. FALSI Flow Diagram

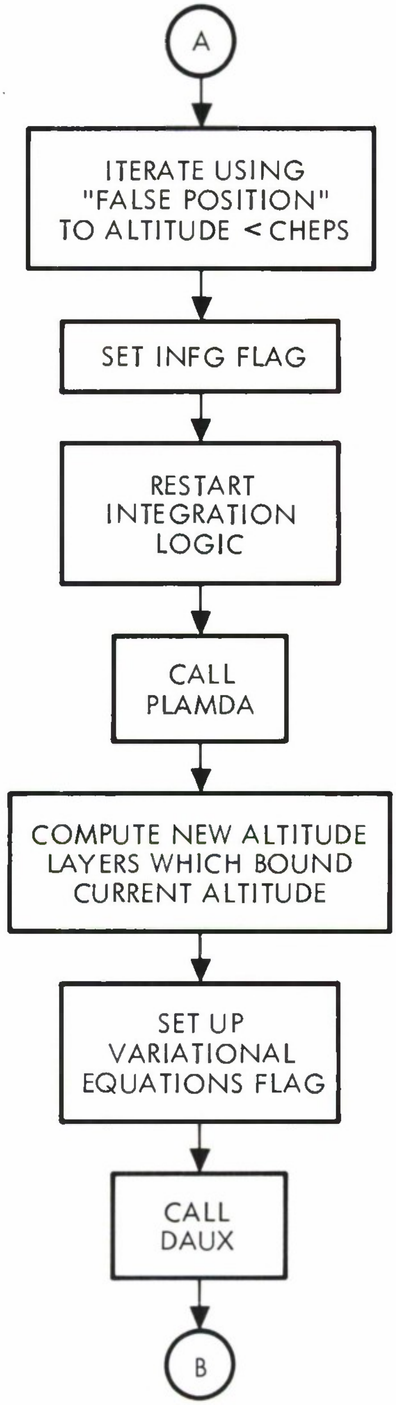


Figure 5-6. FALSI Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title

FIT

B. Segment

NRTPOD

C. Called by subroutine

DCITER

FUNCTION

This subroutine monitors the flow of information through the following sequence of events.

a) Determines whether the current iteration is converging or diverging

b) Forming the solution vector of the differential correction and applying it to give new estimates of the parameters being solved for

c) Test for maximum iterations

d) Set the bounds for the next iteration

e) Test whether 4 solutions in a row have failed to converge

USAGE

A. Calling sequence

CALL FIT

B. Input

1. COMMON

CFTEPS	Convergence criterion, (nominally set to 1.0 E-3).
KOUT	Symbolic output tape (print).
NDPAR1	Identifier showing the starting location of where the solution vector will be stored in variable storage.
NPR	Total number of parameters to solve for.
NITER	Maximum allowable iterations.
IFIT	Identifies predicted RMS's if bounds are used in forming solutions.
CFLAG	Suppresses RMS test when impact has occurred.

NITCT	Counter on number of iterations.
TSUSB	Best RMS so far.
TSUSP	Predicted RMS for next iteration
TZ	Flag to indicate if the solution was affected by the bounds. If the flag is non-zero the solution was affected by the bounds.
XBSQ	Scale factor for BNDS to cause subsequent solutions to be affected by bounds.
TCRASH	Flag to indicate impact, TCRASH $\neq 0$, indicates impact has occurred.
IFTEX	Indicates mode of exit from FIT.
POBCNT	Number of observables actually included (after editing, etc.) on any iteration.
TSUS	Current RMS.
MPR	Size of constrained system.
MBNDS	Variable storage pointer for bounds corresponding to constrained system.
NST	Variable storage pointer for temporary storage used for linear constraints.

2. Calling sequence

-

C. Output

1. COMMON

VSTR (NBDNS)	Array in variable storage containing the set of bounds to be used on the next iteration.

2. Calling sequence

-

D. Error/action messages

"*************MAJOR PROGRAM ERROR....POSSIBLE INPUT AND/OR MACHINE ERROR"

This message is printed if FIT is less than or equal to zero.

SUBROUTINES USED

A. Library

SQRT
ABS

B. Program

BOUNDS	Scale bounds with a given scale factor
LEGS2	Least square package, solves Ax = B
APPLY	Applies differential correction solution vector and prints the iteration summary.

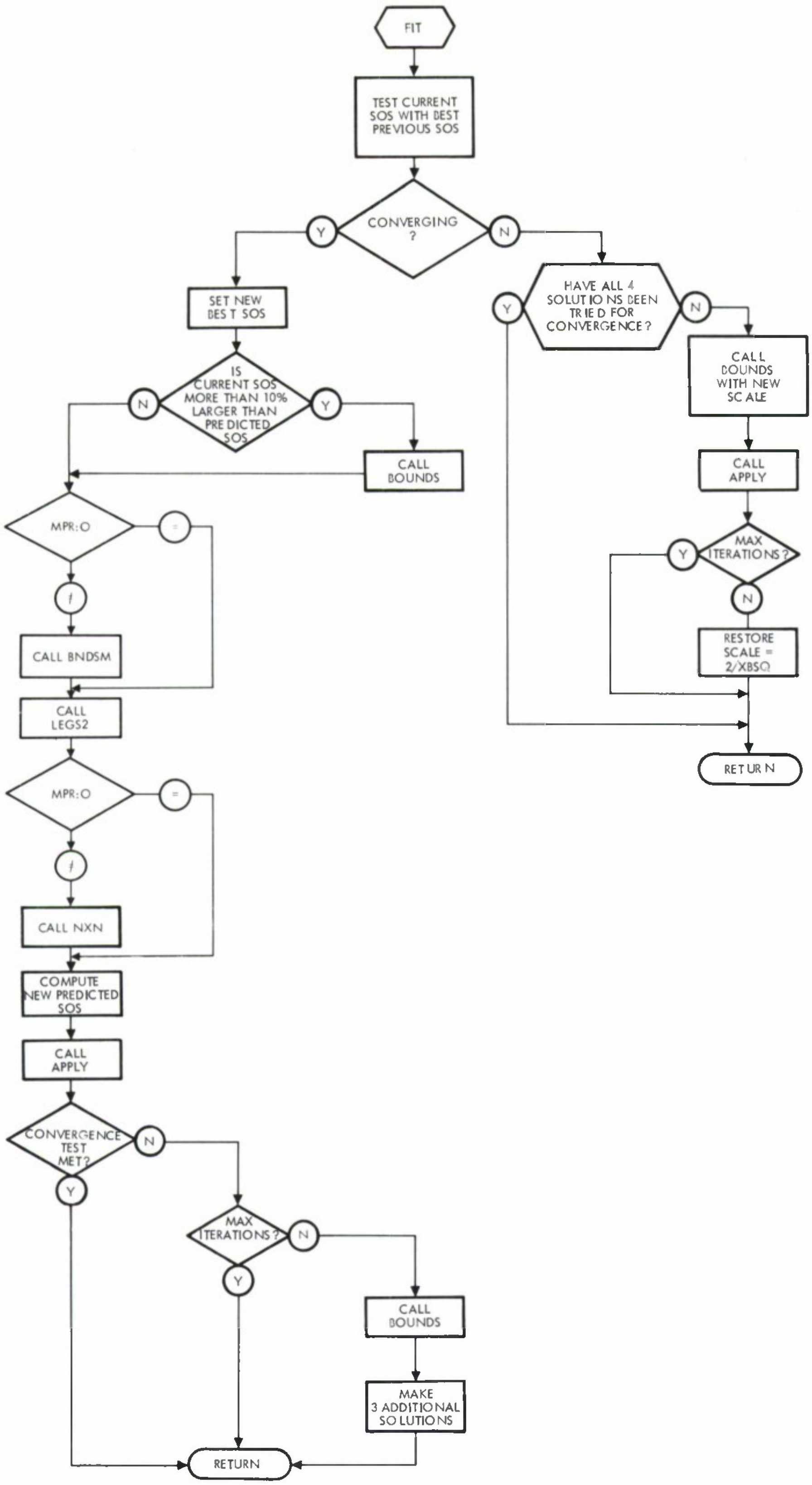


Figure 5-7. FIT Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title
FSIGMA

B. Segment
NRTPOD

C. Called by subroutine
RADR

FUNCTION

Computes the specified functional standard deviation as a function of the nominal standard deviation, the topocentric range, the radar cross section, and other sensor dependent constants.

USAGE

A. Calling sequence
Call FSIGMA (I)

B. Input

1. COMMON

KOUT	Output tape number
CMTER	Meters/e. r.
CDEG	Degrees/radian
CKMER	Kilometers/e. r.
NSTAT	Starting location in VSTR of the master sensor table
NSSTB	Starting location in VSTR of sensor mean and RMS table
NMSTAT	Number of cells in VSTR allotted per sensor
PUBS	Observation vector ID, time, R, A, E, $\dot{R}$, type or ID, time, $\ddot{R}$, 0, 0, 0, type
PCMR	Computed slant range of spacecraft relative to the current station in PSTAT
PUI	The topocentric direction cosines of vehicle position in the equatorial system

PWDTI — Geocentric earth fixed velocity of the vehicle in a station meridian equatorial system.

2. Calling sequence

   I Flag denoting observation tape

   I = 1 denotes range or range acceleration observation type

   = 2 denotes azimuth observation type

   = 3 denotes elevation observation type

   = 4 denotes range rate observation type

C. Output

1. COMMON

   PSIG - Functional standard deviation

2. Calling sequence

   —

D. Error/action messages

If the station ID is not found in the master sensor table, an error comment "ROUTINE FSIGMA CANNOT FIND STATION XXX, INPUT ERROR" is printed off line and the program halts.

SUBROUTINES USED

A. Library

ACOS
EXIT
SQRT

B. Program

DOT	Dot product routine
RADSQ	Magnitude of a vector
FTHET	Computes $f(\theta)$ which is the radar cross section.

EQUATIONS

The specified functional standard deviation, σ_f, is as follows:

$$\sigma_f = \sqrt{A_i^2 + B_i R^4 / f(\theta)}$$

A_i = Nominal standard deviation

B_i = Sensor dependent constant ($i = 1, \ldots, 5$) denoting A, E, R, $\dot{R}$, $\ddot{R}$

R = Topocentric range

$f(\theta)$ = Radar cross section

The function $f(\theta)$ is obtained by linear interpolation from the input table $\left[\theta_j, f(\theta)_j\right]$ $j = 1, \ldots, n$ where $n \leq 7$, each table being sensor dependent. The argument θ is the angle between the drag velocity vector and the topocentric range vector. This angle is sensor dependent and its formulation is derived in Reference 4.

SUBROUTINE IDENTIFICATION

A. Title

FTHET

B. Segment

NRTPOD

C. Called by subroutine

FSIGMA

FUNCTION

FTHET computes f(θ), the radar cross section, by linearly interpolating a sensor specific input table $\left[\theta_j, f(\theta)_j\right]$ j = 1, ..., n, where n ≤ 7. The angle between the drag velocity vector and the topocentric range vector, θ, is given as an argument to FTHET.

USAGE

A. Calling sequence

F = FTHET (J, THETA) (FUNCTION SUBPROGRAM)

B. Input

1. COMMON

VSTR - Variable storage

2. Calling sequence

THETA	angle between the drag velocity vector and the topocentric range vector. Using THETA as an argument, f(θ) is obtained by linear interpolation from the input table $\left[\theta_j, f(\theta_j)\right]$ j = 1, ..., n; where n ≤ 7
J	Integer identifying the position in the master sensor table of the current station ID

C. Output

1. COMMON

—

2. Calling sequence (FUNCTION SUBPROGRAM)

F - Interpolated value of the radar cross section, $f(\theta)$

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

Given the argument θ where $\theta_j \le \theta \le \theta_{j+1}$, the corresponding value of $f(\theta)$ is obtained by linearly interpolating between θ_j, $f(\theta_j)$ and θ_{j+1}, $f(\theta_{j+1})$.

The function, $f(\theta)$, is assumed constant between 0 and θ_1, and between θ_n and π. In other words,

$$\text{if } 0 \le \theta \le \theta_1, \qquad f(\theta) = f(\theta_1)$$

$$\text{if } \theta_n \le \theta \le \pi, \qquad f(\theta) = f(\theta_n)$$

The two following equations will apply when there is either one entry or none in the $[\theta, f(\theta)]$ table.

$$\text{if } n = 1, \qquad f(\theta) = f(\theta_1) \qquad 0 \le \theta \le \pi$$

$$\text{if } n = 0, \qquad f(\theta) = 1 \qquad 0 \le \theta \le \pi$$

SUBROUTINE IDENTIFICATION

A. Title

FVE

B. Segment

NRTPOD

C. Called by subroutines

FALSI
SETIC

FUNCTION

Determines the flags which indicate the $C_D A/m$'s to be solved while in a particular altitude region.

USAGE

A. Calling sequence

Call FVE

B. Input

1. COMMON

VSTR	Variable storage
NLAMS	Number of $C_D A/m$'s in the solution vector
NH	Pointer to location in variable storage where the altitude - $C_D A/m$ table is stored
NLID	Pointer to location in variable storage where the identifiers for the $C_D A/m$'s appearing in the solution vector are stored
NICPR	Number of ADBARV variables in the solution vector
NH1	Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence
NH2	Pointer to location in variable storage of 2nd altitude layer bounding the current region of influence

2. Calling sequence

—

C. Output

1. COMMON

IFVE(1)	Flag indicating whether the 1st C_DA/m of a particular region is in the solution vector
IFVE(2)	Flag indicating whether the 2nd C_DA/m of a particular region is in the solution vector
	IFVE = 0 the C_DA/m of a particular region is not in the solution vector.
	$\neq$ 0 the C_DA/m of a particular region is in the solution vector.
NDPRT	Number of CAT1 variables plus number of (C_DA/m) drag parameters being integrated at any one time (either 6 or 8).

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

HEIGHT

B. Segment

NRTPOD

C. Called by subroutines

DRAG
FALSI

FUNCTION

Computes the altitude of a reference vehicle above the earth where the earth equatorial radius is computed as a function of the geocentric latitude.

USAGE

A. Calling sequence

H = HEIGHT(X) (FUNCTION SUBPROGRAM)

B. Input

1. COMMON

TRM1	See CONST subroutine
TRM2	See CONST subroutine

2. Calling sequence

X - Beginning address of the position vector (earth radii)

C. Output

1. COMMON

—

2. Calling sequence (FUNCTION SUBPROGRAM)

H - Altitude above earth (earth radii)

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRT
EXP

B. Program

—

EQUATIONS

The altitude (h) above the earth is computed as follows:

$$h = r - \rho$$

where r is the radius magnitude of the vehicle and ρ is the radius of the earth at sea level.

$$\rho \approx a_e \left[1 - \left(\frac{k}{2} - \frac{45}{256} k^3 \right) s^2 + \left(\frac{3}{8} k^2 - \frac{15}{32} k^3 \right) s^4 \right]$$

where

$s = \sin \phi$ $\quad\phi$ = geocentric latitude

$k = \frac{2 - \epsilon}{(1 - \epsilon)^2} \epsilon$, ϵ is the ellipticity of the earth

a_e = the earth equatorial radius

SUBROUTINE IDENTIFICATION

A. Title

HINT

B. Segment

NRTPOD

C. Called by subroutines

SETIC
FALSI

FUNCTION

Computes the current two altitude layers that the vehicle lies between during the trajectory simulation.

USAGE

A. Calling sequence

Call HINT (H, ALT)

B. Input

1. COMMON

VSTR	Variable storage
NH	Pointer to location in variable storage where the altitude - C_DA/m table is stored
NLAM	Total number of entries in the altitude (C_DA/m) table
NPXLM	Pointer to location in variable storage where the $\dfrac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial\left(\dfrac{C_DA}{m}\right)_i}$ matrix is stored, $i = 1, \ldots, \text{NLAM}$

NPALM — Pointer to location in variable storage where the b_i's are stored.

$$b_i = \underbrace{\left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial(\alpha, \delta, \beta, A, r, v)_{t=o}}\right]^{-1}}_{6 \times 6} \underbrace{\left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial\left(\frac{C_D A}{m}\right)_i}\right]}_{6 \times 1}$$

$i = 1, \ldots, NLAM$

KOUT — Output tape number

INFG — Flag indicating whether an altitude crossing has occurred and which region of drag influence has been entered

INFG = 0 no altitude crossing occurred

= 1 vehicle has reentered altitude region 1

= 2 vehicle has left influence of altitude region 1 and crossed into altitude region 2

2. Calling sequence

H - current altitude of vehicle (earth radii)

C. Output

1. COMMON

NH1 — Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence

NH2 — Pointer to location in variable storage of the 2nd altitude layer bounding the current region of influence

2. Calling sequence

ALT(I) — Two altitude layers bounding the current region of influence I = 1, 2

D. Error/action messages

If the current altitude is not bounded by the input altitudes the following error message appears off-line: "ALTITUDE = XXXXXX NOT BOUNDED BY INPUT ALTITUDES" and the program halts.

SUBROUTINES USED

A. Library

EXIT

B. Program

—

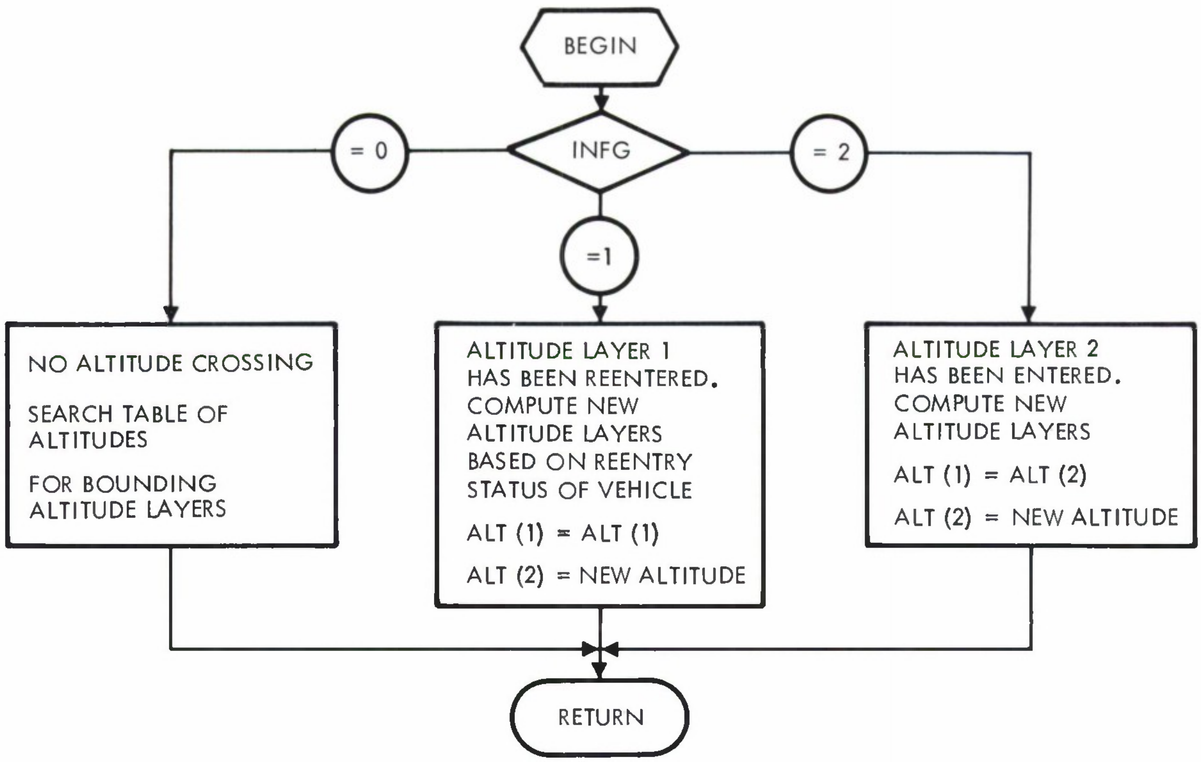


Figure 5-8. HINT Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title
   INPUT

B. Segment
   NRTPOD - Input processor

C. Called by subroutines
   NRTPOD (DRIVER)

FUNCTION

INPUT's function is to serve as a main driver for the Input Processor Link. It utilizes routines to initialize COMMON storage, process NAMELIST input, assign variable storage, and process sensor information and observations.

USAGE

A. Calling sequence
   CALL INPUT

B. Input
   1. //COMMON
      -
   2. Calling sequence
      -

C. Output
   1. //COMMON
      -
   2. Calling sequence
      -

D. Error/action messages
   -

E. Internal storage

1. //COMMON

NDAYS	Number of days of ephemeris data (positions of the moon and sun) accepted on input
DVEHN	Array of 3 BCD words identifying the vehicle number and name (Input to columns 4-17 on JDC card)
DHEAD	2 BCD words containing arbitrary header information. (Input to columns 18-29 on JDC card)
PREFLG	NRTPOD control flags (columns 31-40 on JDC card)
DCFLG	NRTPOD control flags (columns 41-50 on JDC card)
PSTFLG	NRTPOD control flags (columns 51-60 on JDC card)
KIN	Symbolic input tape number
KOUT	Symbolic output tape number
COMLST	Contains size of variable storage

2. Labeled COMMON

/VSTR/	
VSTR	Variable storage array
/INPP/	
DTMP	Temporary cells containing sensor information used by the Input Processor Link
DATA	Temporary cells used only by the Input Processor Link
/EPHCOM/	
ECOM	Array of storage containing the position ephemerides of the moon and sun (Input to NRTPOD)

SUBROUTINES USED

A. Library

-

B. Program

SETCON	Sets up program constants.
RDDATA	Routine to read NAMELIST input and ephemeris data.
ASSIGN	Establishes storage assignments for VSTR (variable storage) arrays.

SETTAB	Sets up VSTR (NIDP), VSTR (NPRCD), VSTR (NPBIS), VSTR (NSCALE), VSTR (NBDNS), and DTMP tables.
SDELET	Moves observation deletion numbers from DATA storage to VSTR (NIDLED).
STSMAT	Convert the upper triangular S matrix in DATA storage from human units to machine units and then transfer to VSTR (NATA).
SUPMAT	Move the initial update matrix from DATA storage to VSTR (NR) and convert from human units to machine units.
DPRLM	Sets up preliminary information for the input processor. This information concerns epoch time and mode of epoch position and velocity.
PRECES	Precess ephemeris data from mean equator and equinox of 1950. 0 to the equator and true equinox of date.
DPROS	Issue calls on the sensor and observation loading routines if required.
PRCONS	Prints program constants, input data, variable storage pointers, and working storage cells.
BIJC	Processes linear constraint data.

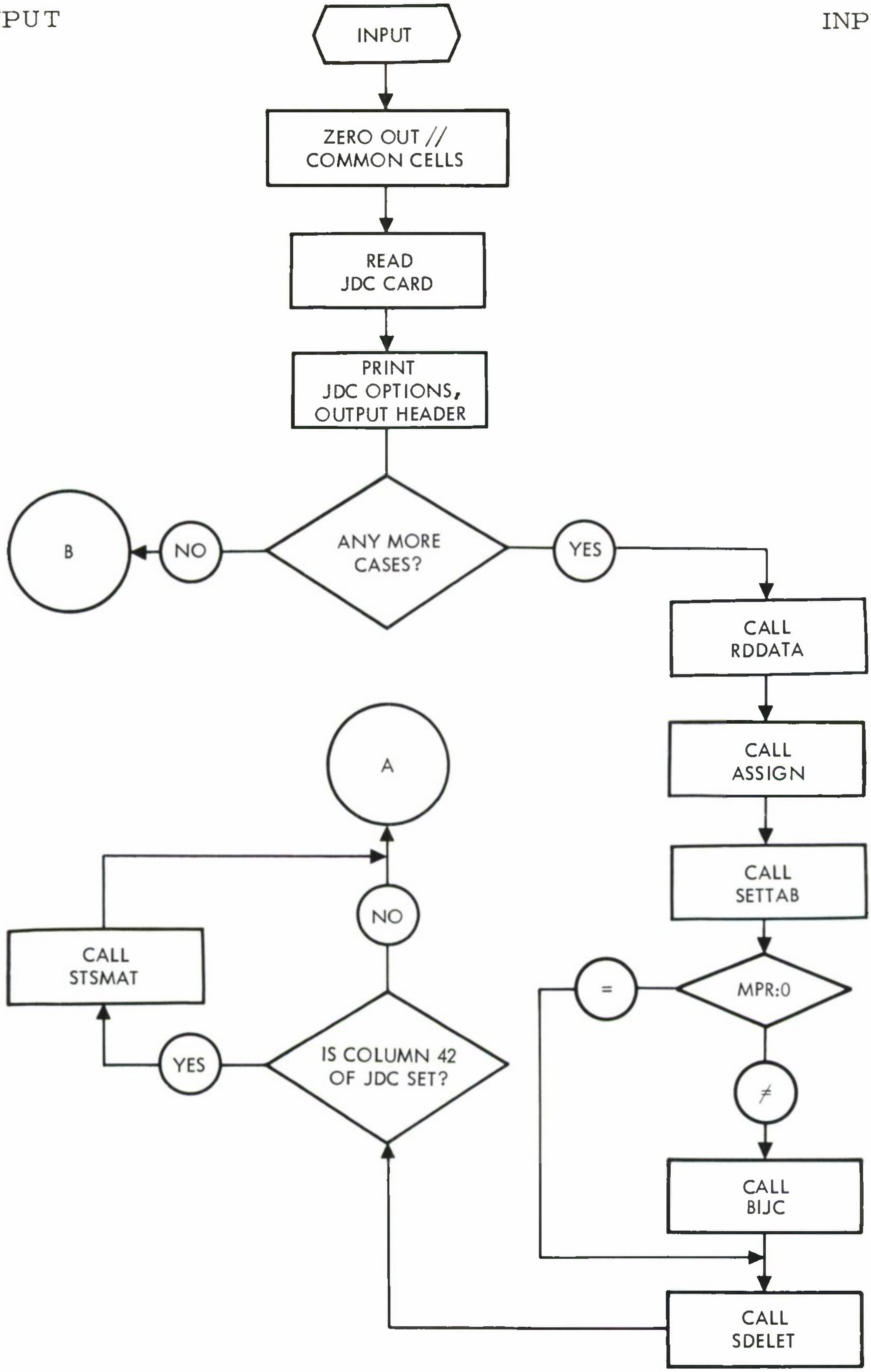


Figure 5-9. Input Flow Diagram

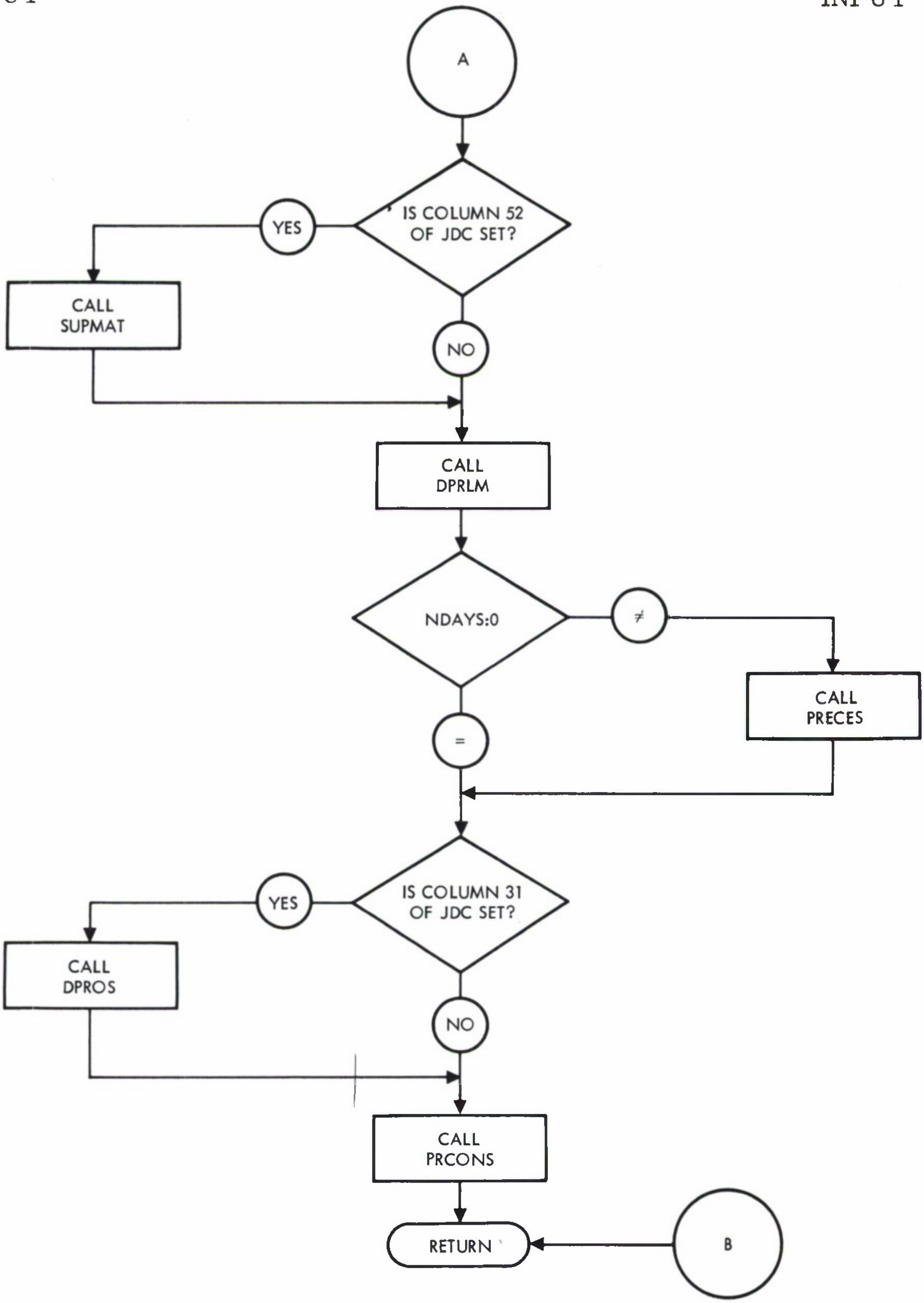


Figure 5-9. Input Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title
JDCPRT

B. Segment
NRTPOD

C. Called by subroutine
INPUT

FUNCTION

Prints and describes the Job Description Card options which have been requested on a particular NRTPOD case.

USAGE

A. Calling sequence
Call JDCPRT

B. Input

1. COMMON

PREFLG	Columns 31-40 of the JDC
DCFLG	Columns 41-50 of the JDC
PSTFLG	Columns 51-60 of the JDC
KOUT	Output tape number

2. Calling sequence

C. Output

1. COMMON

—

2. Calling sequence

—

3. Other

Printed description of the options requested on the JDC

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

LEGS2

B. Segment

NRTPOD

C. Called by subroutine

FIT
PRAUPD

FUNCTIONS

a) To solve an overdetermined linear system of equations Ax = b

b) To compute the inverse of $A^T A$

c) After solving for x, to compute $||Ax - b||^2$

USAGE

A. Calling sequence

Call LEGS2 (NDPAR, Z, SUSP, I1, I2, I4)

B. Input

1. COMMON

NATA	Identifies the starting location of where the upper triangular $A^T A$ is stored
NBDNS	Identifies the starting location for the bounds used by LEGS2
NPR	Number of all parameters to solve for
NR	Identifies the starting location of where the inverse $A^T A$ (in triangular form) is stored

2. Calling sequence

NDPAR	The index for variable storage where the solution vector x is to be stored
I1, I2, I4	Option control flags
NEQ	Size of system
NBD	Pointer in VSTR for system bounds

C. Output

1. COMMON

VSTR (NDPAR)	Start of the array containing the solution vector x
VSTR (NR)	Start of an array containing $(A^TA)^{-1}$ as a lower triangular matrix

2. Calling sequence

Z	Flag to indicate if the solution was affected by the bounds. If the flag is non-zero the solution was affected by the bounds
B	Predicted SOS for the next iteration

SUBROUTINES USED

A. Library

—

B. Programs

—

EQUATIONS

To solve for differential corrections, find x so that $\|Ax - b\|^2$ is minimum under the side condition that

$$\sum_i \left(\frac{x_i}{B_i}\right)^2 \leq 1 \qquad B_1, B_2, \cdots, = \text{bounds}$$

The side condition may be described as

$$x^T B^{-2} x \leq 1$$

where

$$\begin{bmatrix} B_1^{-2} & 0 \cdots & & \\ 0 & B_2^{-2} & & \\ \cdot & & \cdot & \\ \cdot & & & \cdot \end{bmatrix} = B^{-2}$$

B^{-2} is a diagonal matrix

Bounds

Define x(z) as the solution of the linear system

$$(A^T A + zB^{-2})\,X = A^T b$$

where B^{-1} is the diagonal matrix with the (i, i) diagonal element being B_i^{-1} if $B_i > 0$ and $B_i < 0$. If $B_i = 0$, the i^{th} row and column of the augmented normal matrix is ignored and x_i is set to zero.

a) The routine finds $x = x(0)$. If $(B^{-2}x, x) \leq 1 + \epsilon_1$ the solution is obtained. Otherwise

b) Define $y(z) = \left[B^{-2}x(z), x(z)\right]$. Now $y(0) > 1 + \epsilon_1$. Compare y(h), y(1 0h), y(100h), . . . , until a value of z is found with $1 - \epsilon_2 \leq y(z) \leq 1 + \epsilon_1$, in which case x(z) is the solution or until two values of z are found with $y(z_1) > 1 + \epsilon_1$ and $y(z_2) < 1 - \epsilon_2$. The required value of z is now bracketed. Then

c) Choose a value z_3 between z_1 and z_2. If $1 - \epsilon_2 \leq y(z_3) \leq 1 + \epsilon_2$, then $y(z_3)$ is the solution. Otherwise

d) Use inverse quadratic interpolation (to zero) to obtain a new guess z_4. If $1 - \epsilon_2 \leq y(z_4) \leq 1 + \epsilon_1$, then $x(z_4)$ is the solution. Otherwise

e) Select from the set z_1, z_2, z_3, z_4 the two values of z which bracket the solution most tightly. Use these values as z_1 and z_2 and go back to 3.

The iterative process will stop if the number of solutions of the linear system reaches 20.

Linear System

Let $C = A^T A + zB^{-2}$. The routine finds a matrix S with $SCS^T = D$. S is lower triangular with (-1) on the diagonal. It is easy to find S and D for a 1 x 1 matrix C. Assume S and D have been found for a k x k matrix C. Now augment C by another row and column

$$\begin{pmatrix} C & d \\ d^T & a \end{pmatrix}$$

A vector ω and a scalar β are now desired such that

$$\begin{pmatrix} S & 0 \\ \omega^T & -1 \end{pmatrix} \begin{pmatrix} C & d \\ d^T & a \end{pmatrix} \begin{pmatrix} S^T & \omega \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} D & 0 \\ 0 & \beta \end{pmatrix}$$

The requirements are satisfied by

$$\omega = S^T D^{-1} Sd$$

$$\beta = a - \omega^T d$$

The routine builds the matrix S by the above process with $k = 2, 3, \cdots, N$. The final result is a decomposition of the augmented matrix

$$\begin{pmatrix} S & 0 \\ \omega^T & -1 \end{pmatrix} \begin{pmatrix} A^T A + zB^{-2} & A^T b \\ b^T A & b^T b \end{pmatrix} \begin{pmatrix} S^T & \omega \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} D & 0 \\ 0 & a \end{pmatrix}$$

and the N-dimensional vector ω which appears above is the solution vector.

Predicted RMS for Next Iteration

Given $b^T b$, $A^T A$, $A^T b$, X, n = total number of observations

$$\text{Predicted RMS} = \frac{1}{\sqrt{n}} \sqrt{b^T b - 2\, x^T (A^T b) + x^T (A^T A x)}$$

SUBROUTINE IDENTIFICATION

A. Title
LESK

B. Segment
NRTPOD

C. Called by subroutine
PLAMDA

FUNCTION

This subroutine solves the matrix equation AX = B. A is an N x N coefficient matrix, B is an N x M matrix, and X is the N x M solution matrix. Gaussian elimination is used with row interchange taking place to position maximum pivot elements after the rows are initally scaled.

USAGE

A. Calling sequence

Call LESK (C, X, S, N1, M1, N1X, DET, LA)

B. Input

1. COMMON

—

2. Calling sequence

C	The augmented matrix [A, B] of maximal dimension N1X x (N1X + M1X)
N1	The actual order of A
M1	The actual number of columns in B
N1X	The maximal order of A
DET	Both the determinant flag and determinant value if desired

on input DET = $\begin{cases} 0. & \text{No determinant} \\ 1. & \text{Determinant} \end{cases}$

on output DET = 1. on input

C. Output

1. COMMON

—

2. Calling sequence

X	The solution matrix of maximal dimension N1X x M1X
S	A vector of dimension N1X used to store the row scale factors
LA	An error return

LA = +1	during scaling a zero row was found
LA = 0	normal return
LA = -1	$\lvert A(J,J) \rvert < 10^{-10}$ for some J

D. Error/action messages

—

SUBROUTINES USED

A. Library

DABS	Takes absolute value

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title
MATMLT

B. Segment
NRTPOD

C. Called by subroutine
PLAMDA

FUNCTION

Forms the matrix product [A] [B] and stores the result in [C]. [A], [B], and [C] are all singly subscripted arrays stored by rows.

USAGE

A. Calling sequence
Call MATMLT (A, B, C, I, J, K)

B. Input

1. COMMON

—

2. Calling sequence

A - Beginning location of matrix [A]
B - Beginning location of matrix [B]
I, J, K - Dimensions of matrix [A] is I rows by J columns. Dimensions of matrix [B] is J rows by K columns. Dimensions of matrix [C] is I rows by K columns.

C. Output

1. COMMON

—

2. Calling sequence

C - Beginning location of where matrix product [A] [B] is stored

3. Internal Storage

TEMP	Temporary storage used for accumulating the row by row product. TEMP is currently dimensioned (10) in this subroutine which restricts the maximum size of the matrix product to a 10 x 10. This maximum may be modified by merely altering the dimension size of TEMP.

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

MTOC

B. Segment

NRTPOD

C. Called by subroutine
DPRLM

FUNCTION

To update a set of Smithsonian mean elements, convert to osculating and then to Cartesian. It also calls JTOC to convert the Julian date to calendar date.

USAGE

A. Calling sequence

Call MTOC (TNOMX, SMELM, DELT)

B. Input

1. COMMON

DAYINT	Integer portion of Julian date
DAYFRC	Fractional portion of Julian date
CJ2	J2 earth harmonic
C2PI	2π radians
CPI	π radians
KOUT	Output tape unit
CMU	μ ER^3/min^2
CKMER	Conversion from kilometers to earth radii

2. Calling sequence

SMELM	21-word vector containing the Smithsonian mean elements and their time derivatives for updating and conversion to osculating. See Table I.

Table I

Location	Element	Units
SMELM (1)	a	earth radii
(2)	e	--
(3)	i	radians
(4)	Ω	radians
(5)	ω	radians
(6)	M	radians
(7)	$\dot{a}$	er/day
(8)	$\dot{e}$	--/day
(9)	$\dot{i}$	rad/day
(10)	$\dot{\Omega}$	rad/day
(11)	$\dot{\omega}$	rad/day
(12)	n	rad/day
(13)	$\ddot{a}/2$	er/day^2
(14)	$\ddot{e}/2$	--/day^2
(15)	--	--
(16)	$\ddot{\Omega}/2$	rad/day^2
(17)	$\ddot{\omega}/2$	rad/day^2
(18)	$\dot{n}/2$	rad/day^2
(19)	$\ddot{n}/6$	rad/day^3
(20)	$\dddot{n}/24$	rad/day^4

DELT Time to epoch in days, should be greater than 10^{-8} or else set to zero identically

C. Output

1. COMMON

DYEAR Calendar year - 1900
DMNTH Calendar month

DDAY	Calendar day	MCOM (54)
DHOUR	Hour	MCOM (55)
DMIN	Minute	MCOM (56)
DSEC	Second	MCOM (57)

2. Calling sequence

TNOMX	6-word vector containing x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ in kilometers and kilometers/second
TNOMX(1)	x kilometers
TNOMX(2)	y kilometers
TNOMX(3)	z kilometers
TNOMX(4)	$\dot{x}$ kilometers/second
TNOMX(5)	$\dot{y}$ kilometers/second
TNOMX(6)	$\dot{z}$ kilometers/second

D. Error/action messages

E FAILED TO CONVERGE
THE VALUE OF E IS______E___, THE FLAG IS________

This message occurs if the iteration for E has failed to converge after 50 iterations. The flag = 0 indicates the iteration failed for conversion to osculating of the mean elements. The flag = 1 indicates the iteration failed for conversion to Cartesian of the osculating. The program proceeds normally, using the last value of E computed.

SUBROUTINES USED

A. Library

ABS

SIN

COS

ATNQ

SQRT

B. Program

PIMOD	Takes principal value of angle between 0 and 2π

DLSTV	Computes the differentials used in converting from mean to osculating and osculating to mean
JTOC	Converts Julian date to calendar date

EQUATIONS

Given $a_{m_{K-25}}$, e_m, i_m, Ω_m, ω_m, M_m

1. Compute E using

$$E_1 = \pi$$

$$E_{n+1} = E_n + \frac{M_m - E_n + e_m \sin E_n}{1 - e_m \cos E_n}$$

2. Compute true anomaly, v

$$\cos v = \frac{\cos E - e_m}{1 - e_m \cos E}$$

$$\sin v = \frac{\sqrt{1 - e_m^2}\,\sin E}{1 - e_m \cos E}$$

3. Compute radius vector

$$r = a_{m_{K-25}} (1 - e_m \cos E)$$

4. Compute orbital semi-parameter

$$p_m = a_{m_{K-25}} (1 - e_m^2)$$

5. Obtain δ's from DLSTV

6. Compute a_m

$$a_m = \frac{a_{m_{K-25}}}{\left[1 - \frac{A_2}{p_m^2}\left(1 - \frac{3}{2}\sin^2 i_m\right)\sqrt{1 - e_m^2}\right]}$$

7. Compute osculating elements

$$a_{os} = a_m + \delta_{a_m}(a_{m_{K-25}}, e_m, i_m, \Omega_m, \omega_m, M_m)$$

$$e_{os} = e_m + \delta_{e_m}(a_{m_{K-25}}, e_m, i_m, \Omega_m, \omega_m, M_m)$$

$$i_{os} = i_m + \delta_{i_m}(a_{m_{K-25}}, e_m, i_m, \Omega_m, \omega_m, M_m)$$

$$\Omega_{os} = \Omega_m + \delta_{\Omega_m}(a_{m_{K-25}}, e_m, i_m, \Omega_m, \omega_m, M_m)$$

$$\omega_{os} = \omega_m + \delta_{\omega_m}(a_{m_{K-25}}, e_m, i_m, \Omega_m, \omega_m, M_m)$$

$$M_{os} = M_m + \delta_{M_m}(a_{m_{K-25}}, e_m, i_m, \Omega_m, \omega_m, M_m)$$

8. Convert to Cartesian

a. Obtain E and v as above

$$u = v + \omega$$

$$\ell = u + \Omega_{os}$$

$$\ell_r = u - \Omega_{os}$$

b. $U_x = \frac{1}{2}\left[(1 + \cos i_{os}) \cos \ell + (1 - \cos i_{os}) \cos \ell_r\right]$

$U_y = \frac{1}{2}\left[(1 + \cos i_{os}) \sin \ell - (1 - \cos i_{os}) \sin \ell_r\right]$

$U_z = \sin u \sin i_{os}$

$V_x = -\frac{1}{2}\left[(1 + \cos i_{os}) \sin \ell + (1 - \cos i_{os}) \sin \ell_r\right]$

$V_y = \frac{1}{2}\left[(1 + \cos i_{os}) \cos \ell - (1 - \cos i_{os}) \cos \ell_r\right]$

$V_z = \cos u \sin i_{os}$

c. $r = a_{os}(1 - e_{os} \cos E)$

$\dot{r} = \frac{\sqrt{\mu a_{os}}}{r} e_{os} \sin E$

$r\dot{v} = \frac{\sqrt{\mu a_{os}\left(1 - e_{os}^2\right)}}{r}$

d. $x = rU_x$

$y = rU_y$

$z = rU_z$

$\dot{x} = \dot{r}U_x + r\dot{v}V_x$

$\dot{y} = \dot{r}U_y + r\dot{v}V_y$

$\dot{z} = \dot{r}U_z + r\dot{v}V_z$

SUBROUTINE IDENTIFICATION

A. Title
NTOM

B. Segment
NRTPOD

C. Called by subroutine
RADR

FUNCTION

To reduce a row of the unconstrained observational equations to the constrained system.

USAGE

A. Calling sequence
Call NTOM

B. Input

1. COMMON

NPR	Size of unconstrained system
NAROW	NSTR pointer for row of observational equations
NR	VSTR pointer for ATA (constrained)
MPR	Size of constrained system
IMAX	Number of non-zero elements of constraint matrix
NIJ	IVSTR pointer for coded subscripts of constraint matrix
NB	VSTR pointer for non-zero elements of constraint matrix
NC	VSTR pointer for additive constants
CFLG	Flag for additive constants

2. Calling sequence

C. Output

1. COMMON

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

$A_o = AB$ where A_o is the constrained set of observational equations, A the unconstrained set, and B the constraint matrix.

SUBROUTINE IDENTIFICATION

A. Title

NXN

B. Segment

NRTPOD

C. Called by subroutine

FIT

FUNCTION

NXN expands the constrained inverse and the solution vector to the unconstrained system.

USAGE

A. Calling sequence

Call NXN (NDPAR, S, RI)

B. Input

1. COMMON

NPR	Size of unconstrained system
NR	VSTR pointer for ATA (constrained)
MPR	Size of constrained system
IMAX	Number of non-zero elements in constraint matrix
NIJ	IVSTR pointer for coded subscripts of constraint matrix
NST	Temporary storage used in VSTR for constraining the size of the system from NPR to MPR
NB	VSTR pointer for non-zero elements of constraint matrix
NC	VSTR pointer for additive constants
CFLG	Flag for additive constants
VSTR IVSTR	Variable storage

2. Calling sequence

NDPAR	VSTR pointer by solution vector
S	Block of temporary storage [MPR (MPR + 1)/2]
RI	Block of temporary storage MPR

C. Output

1. COMMON

VSTR (NDPAR)	Constrained solution vector
VSTR (NR)	Constrained A^TA

2. Calling sequence

—

SUBROUTINES USED

A. Library

B. Program

ELEM

EQUATIONS

$$x = By + c$$

$$(A^TA)^{-1} = B\left(A_o^T A_o\right)^{-1} B^T$$

where

x = unconstrained solution vector

y = constrained solution vector

B = constraint matrix

$\left(A_o^T A_o\right)^{-1}$ = inverse of constrained normal matrix

SUBROUTINE IDENTIFICATION

A. Title

OBSIN

B. Segment

NRTPOD - Input Processor

C. Called by subroutine

LODOBS

FUNCTION

Function is to apply sensor biases, if any, scale observation data and weights (σ's) to internal units, and move this data from temporary storage to permanent storage (Z). This routine overrides the weights input on sensor cards by the weights, if any, input on the observation cards.

USAGE

A. Calling sequence

Call OBSIN (Z, ISTART, NOB)

B. Input

1. Blank COMMON

CKMER	(km/e.r.)
CDEG	(Deg/radian)
KOUT	Output tape number
NSSTB	VSTR pointer for station means and RMS information
NSTAT	VSTR pointer for master sensor table Julian date of midnight, epoch day

2. Labeled COMMON

/TEMP/

TEMP (1)	Station ID
TEMP (2-7)	Time of observation in year, month, day, hour, minute, second
TEMP (8)	type
TEMP (9)	R, range (e.r.) or $\ddot{R}$, range acceleration (e.r./min^2)
TEMP (10)	A, azimuth (rad)
TEMP (11)	E, elevation (rad)
TEMP (12)	$\dot{R}$, range rate (e.r./min)
TEMP (13)	σ_R, standard deviation in range (e.r.) or $\sigma_{\ddot{R}}$, standard deviation in range acceleration (e.r./min^2)
TEMP (14)	σ_A, standard deviation in azimuth (rad)
TEMP (15)	σ_E, standard deviation in elevation (rad)
TEMP (16)	$\sigma_{\dot{R}}$, standard deviation in range rate (e.r./min)

/INPP/	
NDTMP	Counter on the DTMP buffer for biases and weights by station
DTMP	Buffer storage for station and observation biases along with their respective weights (σ's)
/VSTR/	
VSTR	Variable storage array

3. Calling sequence

ISTART	Starting location of Z

C. Output

1. COMMON

—

2. Calling sequence

Z (ISTART)	STATION ID
Z (ISTART +1)	Time from epoch (min)
Z (ISTART +2)	R, range (e.r.) or $\ddot{R}$, range acceleration (e.r./min^2)
Z (ISTART +3)	A, azimuth (rad)
Z (ISTART +4)	E, elevation (rad)
Z (ISTART +5)	$\dot{R}$, range rate (e.r./min)
Z (ISTART +6)	Type
Z (ISTART +7)	σ_R, standard deviation in range (e.r.) or $\sigma_{\ddot{R}}$, standard deviation in range acceleration (e.r./min)
Z (ISTART +8)	σ_A, standard deviation in azimuth (rad)
Z (ISTART +9)	σ_E, standard deviation in elevation (rad)
Z (ISTART +10)	$\sigma_{\dot{R}}$, standard deviation in range rate (e.r./min)

Note: Whenever $\ddot{R}$ observations are processed, the rest of the Z buffer contains zeros.

NOB	Flag to indicate error in observation ID. =0 ID found in master sensor table. $\neq$0 ID not found in master sensor table.

D. Error/action messages

1. Off-line comment

"ERROR IN OBSERVATION ID _____"

2. On-line comment

—

3. Action

Set NOB flag, return to calling program.

SUBROUTINES USED

A. Library

—

B. Program

TIME - Computes Julian date and minutes from midnight of epoch day

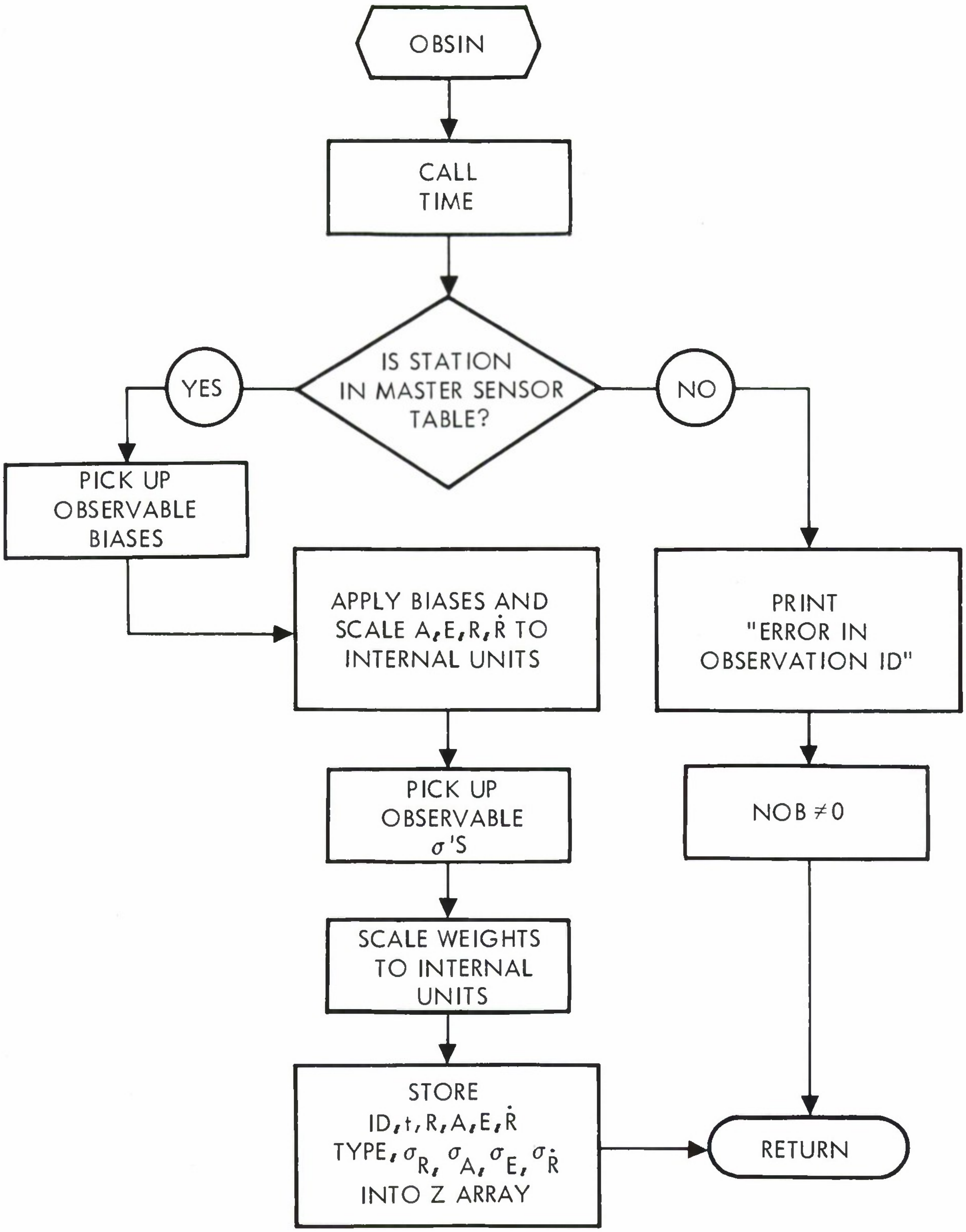


Figure 5-10. OBSIN Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

PAGE2

B. Segment

NRTPOD

C. Called by subroutine

LINES

FUNCTION

Prints out the functional standard deviations on option (JDC column 43).

USAGE

A. Calling sequence

Call PAGE2

B. Input

1. COMMON

KOUT	Output tape number

2. Calling sequence

—

C. Output

1. COMMON

SBUF — Buffer array containing sets of functional standard deviations for each residual to make up one page of functional sigma output.

SBUF(1) - Station ID

(2) - Residual type $= 0(R, A, E, \dot{R},)$
$= 1(\ddot{R})$

(3) - Time of residual (min from epoch)

(4) - Residual count

(5) - σ_R (kilometers) or $\sigma_{\ddot{R}} \left(\frac{\text{meters}}{\text{sec}^2}\right)$

(6) - σ_A (deg.)

(7) - σ_E (deg.)

 PAGE2

(8) - $\sigma_{\dot{R}}$ (km/sec)

(9) - Station ID

.
.
.

(16) - $\sigma_{\dot{R}}$ (km/sec)

(17) - Station ID

.
.
.

(24) - $\sigma_{\dot{R}}$ (km/sec)

etc.

Eight cells per residual until SBUF is filled with an output page of functional standard deviations.

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

PLAMDA

B. Segment

NRTPOD

C. Called by subroutines

FALSI
TRJGEN

FUNCTION

Subroutine PLAMDA computes the partials of x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$, $\ddot{x}$, $\ddot{y}$, $\ddot{z}$, with respect to each λ_i, where the λ_i are the drag parameters $\left(\frac{C_D A}{m}\right)_i$ in the solution vector.

USAGE

A. Calling sequence

Call PLAMDA (IFLAG)

B. Input

1. COMMON

VSTR	Variable storage
PR2DPI(1-3)	$\frac{\partial \ddot{x}}{\partial \alpha}, \frac{\partial \ddot{y}}{\partial \alpha}, \frac{\partial \ddot{z}}{\partial \alpha}$
(4-6)	$\frac{\partial \ddot{x}}{\partial \delta}, \frac{\partial \ddot{y}}{\partial \delta}, \frac{\partial \ddot{z}}{\partial \delta}$
⋮	⋮
(16-18)	$\frac{\partial \ddot{x}}{\partial V}, \frac{\partial \ddot{y}}{\partial V}, \frac{\partial \ddot{z}}{\partial V}$
(19-21)	$\frac{\partial \ddot{x}}{\partial \lambda_j}, \frac{\partial \ddot{y}}{\partial \lambda_j}, \frac{\partial \ddot{z}}{\partial \lambda_j}$
(22-24)	$\frac{\partial \ddot{x}}{\partial \lambda_{j+1}}, \frac{\partial \ddot{y}}{\partial \lambda_{j+1}}, \frac{\partial \ddot{z}}{\partial \lambda_{j+1}}$

Where λ_j, λ_{j+1} are the current $\frac{C_D A}{m}$ drag parameters in the region of influence

KOUT — Output tape number

NPALM — Pointer to variable storage location where the b_i's are stored

$$b_i = \underbrace{\left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial(\alpha, \delta, \beta, A, r, v)_{t=t_o}}\right]^{-1}}_{6 \times 6} \underbrace{\left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial \lambda_i}\right]}_{6 \times 1}$$

$$i = 1, \text{NLAM and } \lambda_i \equiv \left(\frac{C_D A}{m}\right)_i$$

NICPR — Number of ADBARV variables in the solution vector

NPXLM — Pointer to location in variable storage where the

$$\left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial\left(\frac{C_D A}{m}\right)_i}\right] \text{ are stored}$$

$i = 1, \ldots, \text{NLAM}$

NLAM — Total number of entries in the altitude $\left(\frac{C_D A}{m}\right)$ table

NPXDLM — Pointer to location in variable storage where the

$$\left[\frac{\partial(\ddot{x}, \ddot{y}, \ddot{z})}{\partial\left(\frac{C_D A}{m}\right)_i}\right] \text{ are stored}$$

$i = 1, \ldots, \text{NLAM}$

NH — Pointer to location in variable storage where the altitude - $C_D A/m$ table is stored

IFVE — Flags indicating whether the $C_D A/m$ in the region of influence is in the solution vector or not

IFVE = 0	the C_DA/m of a particular region is not in the solution vector
≠ 0	the C_DA/m of a particular region is in the solution vector

INFG — Flag indicating whether an altitude crossing has occurred and which region of drag influence has been entered.

INFG = 0	no altitude crossing has occurred
= 1	vehicle has reentered altitude region 1
= 2	vehicle has left influence of altitude region 1 and crossed into altitude region 2

NDPRT — Number of CAT1 variables plus number of C_DA/m drag parameters being integrated at any one time (either 6 or 8)

NH1 — Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence

NH2 — Pointer to location in variable storage of the 2nd altitude layer bounding the current region of influence

2. Calling sequence

IFLAG — Flag indicating to subroutine PLAMDA whether time is at an observation point or an altitude cutoff point

IFLAG = 0	signifies time at an observation time
= 1	signifies time at altitude cutoff

C. Output

1. COMMON

TRAJX(46-51) $\frac{\partial x}{\partial \lambda_i}, \frac{\partial y}{\partial \lambda_i}, \frac{\partial z}{\partial \lambda_i}, \frac{\partial \dot{x}}{\partial \lambda_i}, \frac{\partial \dot{y}}{\partial \lambda_i}, \frac{\partial \dot{z}}{\partial \lambda_i}$

TRAJX(52-57) $\frac{\partial x}{\partial \lambda_{i+1}}, \frac{\partial y}{\partial \lambda_{i+1}}, \frac{\partial z}{\partial \lambda_{i+1}}, \frac{\partial \dot{x}}{\partial \lambda_{i+1}}, \frac{\partial \dot{y}}{\partial \lambda_{i+1}}, \frac{\partial \dot{z}}{\partial \lambda_{i+1}}$

where λ_i and λ_{i+1} indicate the two (C_DA/m) drag parameters in the region of influence.

2. Calling sequence

—

D. Error/action messages

If difficulty is encountered when computing

$$\left[\frac{\partial (x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial (\alpha, \delta, \beta, A, r, v)_{t=o}}\right]^{-1}$$

where t_i is the time at the i^{th} altitude layer,

Subroutine LESK returns a non-zero error flag. If this occurs an error comment, "LINEAR EQUATION SOLVER - LESK ERROR RETURN, LA =____". is printed and the program halts.

SUBROUTINES USED

A. Library

EXIT

B. Program

LESK
MATMLT
TRJOUT

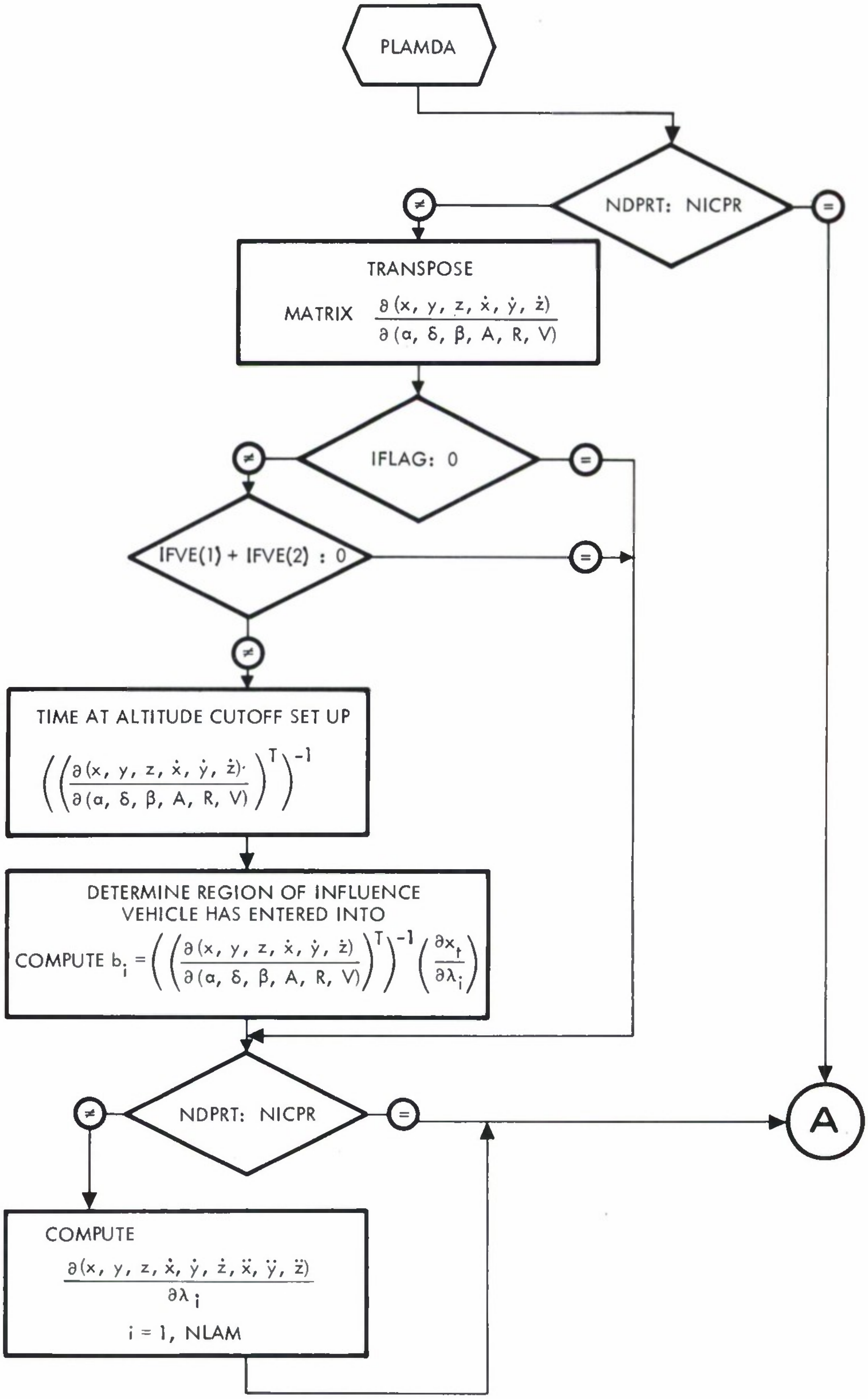


Figure 5-11. PLAMDA Flow Diagram

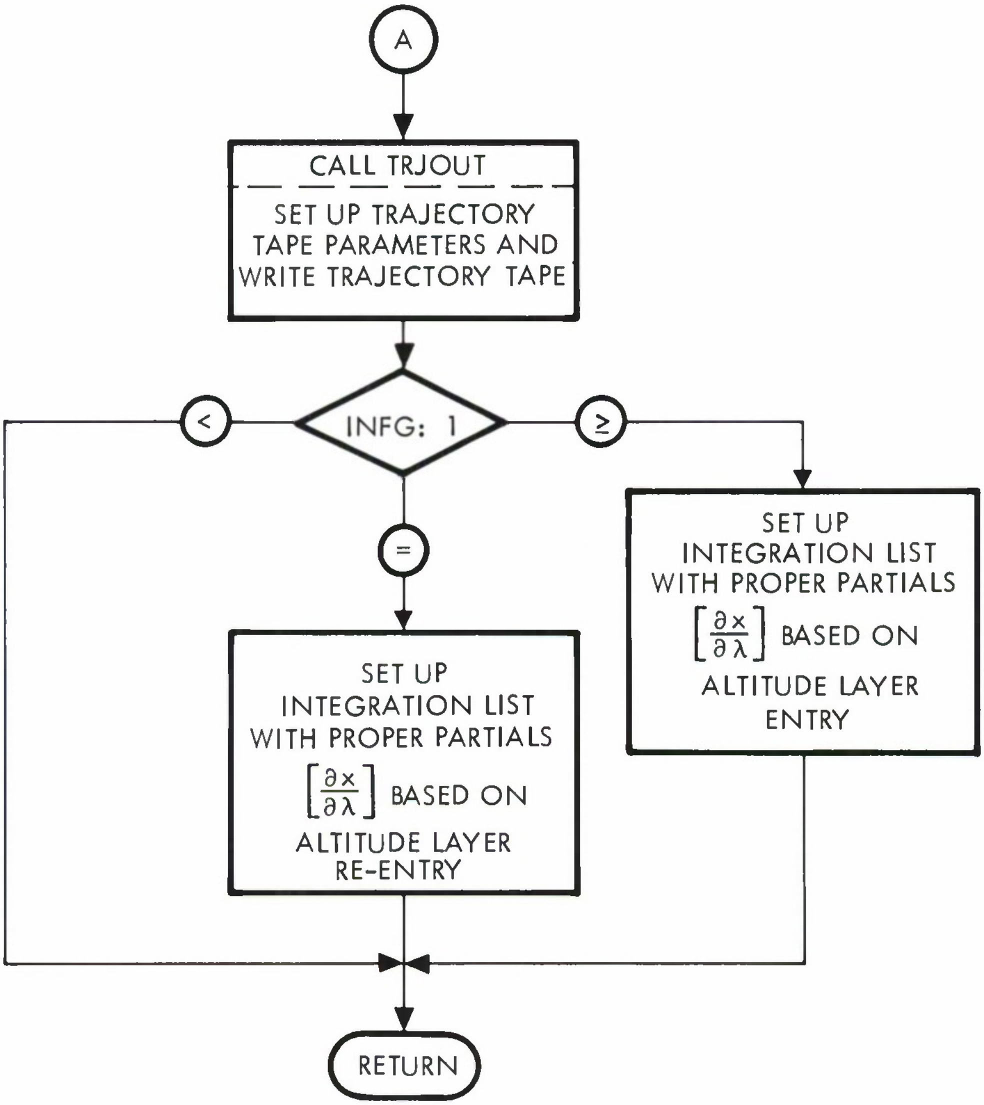


Figure 5-11. PLAMDA Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title
POPPC

B. Segment
NRTPOD

C. Called by subroutine
PRAUPD

FUNCTION

The function is to compute the matrix which takes a Cartesian covariance matrix into an ECI orbit plane matrix up, down, cross. The dimension of the matrix U is the total number of Category 1 variables and drag variables in the solution vector.

USAGE

A. Calling sequence
Call POPPC

B. Input

1. COMMON

NDPR	Total number of Category 1 variables to solve for
TEMP	Temporary storage
TRAJX(1)	x (e. r.)
TRAJX(2)	y (e. r.)
TRAJX(3)	z (e. r.)
TRAJX(4)	$\dot{x}$ (e. r. /min)
TRAJX(5)	$\dot{y}$ (e. r. /min)
TRAJX(6)	$\dot{z}$ (e. r. /min)

2. Calling sequence

C. Output

1. COMMON

TDPDX	Contains matrices of partials for covariance matrix update

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRT

B. Program

—

EQUATIONS

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$$

$$r \cdot v = x\dot{x} + y\dot{y} + z\dot{z}$$

$$|J| = \sqrt{r^2 v^2 - (r \cdot v)^2}$$

$$|J \times r| = \sqrt{r^2 (r^2 v^2 - (r \cdot v)^2)}$$

$$\xi_x = \frac{x}{r} \qquad \xi_y = \frac{y}{r} \qquad \xi_z = \frac{z}{r}$$

$$\eta_x = \frac{r^2 \dot{x} - (r \cdot v)x}{|J \times r|} \qquad \eta_y = \frac{r^2 \dot{y} - (r \cdot v)y}{|J \times r|} \qquad \eta_z = \frac{r^2 \dot{z} - (r \cdot v)z}{|J \times r|}$$

$$\zeta_x = \frac{y\dot{z} - z\dot{y}}{|J|} \qquad \zeta_y = \frac{z\dot{x} - x\dot{z}}{|J|} \qquad \zeta_z = \frac{x\dot{y} - y\dot{x}}{|J|}$$

$$[A] = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}$$

For NDPR = 6

$$TDPDX = U = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}$$

For NDPR > 6

$$TDPDX = U = \begin{array}{ll} \left[\begin{array}{cc|c} A & 0 & 0 \\ 0 & A & 0 \\ \hline 0 & 0 & I \end{array}\right] & \begin{array}{l} \\ \\ \left.\vphantom{I}\right\} NDPR - 6 \end{array} \\ \qquad\qquad\underbrace{\qquad}_{NDPR - 6} & \end{array}$$

where I is the identity matrix

SUBROUTINE IDENTIFICATION

A. Title
PPLPC

B. Segment
NRTPOD

C. Called by subroutine
PRAUPD

FUNCTION

Function is to compute the partial of polar coordinates with respect to Cartesian coordinates and to set up a matrix U necessary to perform the update $V = U \Sigma_x U^T$. The dimension of the matrix U is the total number of Category 1 variables and $C_D A/m$ drag parameters being solved for.

USAGE

A. Calling sequence
Call PPLPC

B. Input

1. COMMON

NDPR	Total number of CAT 1 variables and $C_D A/m$ drag parameters being solved for.
TRAJX	Position, velocity, and acceleration vectors of the vehicle. Also the TRAJX array contains the variational matrix computed from the integration of the variational equations.

C. Output

1. COMMON

TDPDX	Contains the matrices of partials for covariance matrix update (See equations.)

D. Error/action messages

—

SUBROUTINES USED

A. Library

COS	Cosine function
SQRT	Square root function

B. Program

ATNQ	Arc tangent function

EQUATIONS

$$r^2 = x^2 + y^2 + z^2$$

$$v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$$

$$r\dot{r} = x\dot{x} + y\dot{y} + z\dot{z}$$

$$\dot{r} = \frac{r\dot{r}}{r}$$

$$q = \frac{1}{r\sqrt{v^2 - \dot{r}^2}}$$

$$A = \tan^{-1}\left(\frac{x\dot{y} - y\dot{x}}{r\dot{z} - z\dot{r}}\right)$$

$$W = \frac{\cos^2 A}{r\dot{z} - z\dot{r}}$$

$$\frac{\partial\alpha}{\partial x} = \frac{-y}{x^2 + y^2} \quad \frac{\partial\alpha}{\partial y} = \frac{x}{x^2 + y^2} \quad \frac{\partial\alpha}{\partial z} = \frac{\partial\alpha}{\partial\dot{x}} = \frac{\partial\alpha}{\partial\dot{y}} = \frac{\partial\alpha}{\partial\dot{z}} = 0$$

$$\frac{\partial\delta}{\partial x} = \frac{-xz}{r^2\sqrt{x^2 + y^2}} \quad \frac{\partial\delta}{\partial y} = \frac{-yz}{r^2\sqrt{x^2 + y^2}} \quad \frac{\partial\delta}{\partial z} = \frac{\sqrt{x^2 + y^2}}{r^2} \quad \frac{\partial\delta}{\partial\dot{x}} = \frac{\partial\delta}{\partial\dot{y}} = \frac{\partial\delta}{\partial\dot{z}} = 0$$

$$\frac{\partial\beta}{\partial x} = q\left(\frac{x\dot{r}}{r} - \dot{x}\right) \quad \frac{\partial\beta}{\partial y} = q\left(\frac{y\dot{r}}{r} - \dot{y}\right) \quad \frac{\partial\beta}{\partial z} = q\left(\frac{z\dot{r}}{r} - \dot{z}\right)$$

$$\frac{\partial\beta}{\partial\dot{x}} = q\left(\frac{\dot{x}r\dot{r}}{v^2} - x\right) \quad \frac{\partial\beta}{\partial\dot{y}} = q\left(\frac{\dot{y}r\dot{r}}{v^2} - y\right) \quad \frac{\partial\beta}{\partial\dot{z}} = q\left(\frac{\dot{z}r\dot{r}}{v^2} - z\right)$$

$$\frac{\partial A}{\partial x} = W\left[\dot{y} - \frac{\tan A}{r}\left(\dot{z}x - \dot{x}z + \frac{zx\dot{r}}{r}\right)\right]$$

$$\frac{\partial A}{\partial y} = W\left[-\dot{x} - \frac{\tan A}{r}\left(\dot{z}y - \dot{y}z + \frac{zy\dot{r}}{r}\right)\right]$$

$$\frac{\partial A}{\partial z} = \dot{r}\ W \tan A\left(1 - \frac{z^2}{r^2}\right)$$

$$\frac{\partial A}{\partial \dot{x}} = W\left(-y + \frac{xz}{r}\tan A\right)$$

$$\frac{\partial A}{\partial \dot{y}} = W\left(x + \frac{zy}{r}\tan A\right)$$

$$\frac{\partial A}{\partial \dot{z}} = -r\ W \tan A\left(1 - \frac{z^2}{r^2}\right)$$

$$\frac{\partial r}{\partial x} = \frac{x}{r}\quad \frac{\partial r}{\partial y} = \frac{y}{r}\quad \frac{\partial r}{\partial z} = \frac{z}{r}\quad \frac{\partial r}{\partial \dot{x}} = \frac{\partial r}{\partial \dot{y}} = \frac{\partial r}{\partial \dot{z}} = 0$$

$$\frac{\partial v}{\partial x} = \frac{\partial v}{\partial y} = \frac{\partial v}{\partial z} = 0\quad \frac{\partial v}{\partial \dot{x}} = \frac{\dot{x}}{v}\quad \frac{\partial v}{\partial \dot{y}} = \frac{y}{v}\quad \frac{\partial v}{\partial \dot{z}} = \frac{\dot{z}}{v}$$

$$\left[A\right] = \begin{bmatrix} \frac{\partial \alpha}{\partial x} & \frac{\partial \alpha}{\partial y} & \frac{\partial \alpha}{\partial z} & \frac{\partial \alpha}{\partial \dot{x}} & \frac{\partial \alpha}{\partial \dot{y}} & \frac{\partial \alpha}{\partial \dot{z}} \\ \frac{\partial \delta}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial \delta}{\partial \dot{z}} \\ \frac{\partial \beta}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial \beta}{\partial \dot{z}} \\ \frac{\partial A}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial A}{\partial \dot{z}} \\ \frac{\partial r}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial r}{\partial \dot{z}} \\ \frac{\partial V}{\partial x} & \frac{\partial V}{\partial y} & \frac{\partial V}{\partial z} & \frac{\partial V}{\partial \dot{x}} & \frac{\partial V}{\partial \dot{y}} & \frac{\partial V}{\partial \dot{z}} \end{bmatrix}$$

PPLPC

For NDPR = 6

$$TDPDX = U = \begin{bmatrix} A \end{bmatrix}$$

For NDPR > 6

$$TDPDX = U = \left[\begin{array}{c|c} A & 0 \\ \hline 0 & I \end{array}\right]$$

where I is the identity

SUBROUTINE IDENTIFICATION

A. Title

PRAUPD

B. Segment

NRTPOD

C. Called by subroutine

PRUDRV

FUNCTION

Function is to update a given covariance matrix to a specified time t, and to print the resulting matrices. The covariance matrix to be updated can either be a 6x6 (a, δ, β, A, R, v) or an n x n (α, δ, B, A, R, v, C_DA/m'S). The updated normal matrix (polar spherical coordinates) and a correlation matrix is printed. Eigenvalues and eigenvectors of the UVW covariance matrix are output along with determinants of the phase space Cartesian covariance matrix.

USAGE

A. Calling sequence

CALL PRAUPD

B. Input

1. COMMON

KOUT	Symbolic output tape (print)
NPR	Number of all parameters to solve for
NDPR	Number of Category 1 parameters to solve for
NATA	Starting location of where the triangular A^TA is stored [VSTR(NATA)]
NR	Starting location of where the inverse A^TA is stored [VSTR(NR)]
NSCALE	Starting location of the list of conversion factors which convert from machine to output units and vice versa
NDPAR1	Starting location where the solution vector will be stored
NRTMP	Starting location of temporary storage for special handling of the R matrix
NBDNS	Starting location for the bounds used by LEGS2

TEMP	Temporary storage
VSTR	Variable storage. VSTR(NR), VSTR (NRTMP), VSTR (NBDNS) etc.
TRAJX	Contains the position, velocity and acceleration vectors of the vehicle The variational equations may also be present in TRAJX
TZ	Indicates if the solution was affected by bounds
CKMER	Km/e.r.
CDEG	Deg/radian
KOUT	Output tape number
ITRJTP	Trajectory tape number
DYEAR	Epoch year
DMNTH	Epoch month
DDAY	Epoch day
TG	Current integration time
TCRASH	Impact flag =0 no impact ≠ 0 impact

2. Calling sequence

—

C. Output

Off-line print

Sigma and Rho matrix (polar spherical coordinates)

Normal matrix (polar spherical coordinates)

Covariance matrix output (see description of output)

This routine outputs the eigenvalues and eigenvectors along with yaw-roll-pitch rotations aligning the UVW system with the principal axes of error ellipsoid. The determinants mentioned in the output description section are also output by this routine.

D. Error/action messages

SUBROUTINES USED

A. Library

—

B. Program

MATPT
HUMAH
PPLPC
CORMAT
MABAT
LEGS2
POPPC
PRAXIS
UPPER

SUBROUTINE IDENTIFICATION

A. Title

PRAXIS

B. Segment

NRTPOD

C. Called by subroutine

UPDATE

FUNCTION

The functions of this subroutine are described below:

a) To compute the eigenvalues and eigenvectors of a real symmetric 3 x 3 matrix, A (stored as a lower triangular matrix). The eigenvectors for the columns of a matrix U and are ordered as column vectors in such a way that the sum of the diagonal elements of the U matrix is maximized.

b) These eigenvectors are then used to compute the three angles ϕ_1, ϕ_2, ϕ_3 which will resolve the matrix A into a diagonal matrix with the eigenvalues of A as the diagonal elements.

USAGE

A. Calling sequence

Call PRAXIS (A, I, B, J)

B. Input

1. COMMON

—

2. Calling sequence

L(A)	Address of an array A where the matrix is stored
I	Index to indicate just where in the above array the first element of the matrix is [i. e., A(I) is the first element of the matrix].
L(B)	Address of an array B where the results of PRAXIS are to be stored
J	Index to indicate just where in the above array the first element of the results are to be stored (See Output for arrangement of results in array B).

C. Output

1. COMMON

—

2. Calling sequence

B(J) - λ_1 } eigenvalues of A
B(J+1) - λ_2
B(J+2) - λ_3

B(J+3) - U_{11} } first eigenvector
B(J+4) - U_{12}
B(J+5) - U_{13}

B(J+6) - U_{21} } second eigenvector
B(J+7) - U_{22}
B(J+8) - U_{23}

B(J+9) - U_{31} } third eigenvector
B(J+10) - U_{32}
B(J+11) - U_{33}

B(J+12) - ϕ_1 } rotational angles (rad)
B(J+13) - ϕ_2
B(J+14) - ϕ_3

B(J+15) - $\sqrt{\lambda_1}$ } square roots of the three eigenvalues
B(J+16) - $\sqrt{\lambda_2}$
B(J+17) - $\sqrt{\lambda_3}$

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRT
COS
SIN

B. Program

ATNQ	Arctangent routine
XCROSS	Cross product routine

EQUATIONS

Compute the eigenvalues of A

$$m = \frac{1}{3}\,\mathrm{tr}\,(A) \text{ where } \mathrm{tr}(A) = \sum_{i=1}^{3} a_{ii}$$

$$q = \frac{1}{2}\det(A - mI)$$

6p = sum of the squares of the elements of (A - mI). From "Cardano's" trigonometric solution of det $[(A - mI) - \mu I]$ as a cubic in μ, the eigenvalues of A are

$$\lambda_1 = m + 2\sqrt{p}\cos\phi$$

$$\lambda_2 = m - \sqrt{p}\,(\cos\phi + \sqrt{3}\sin\phi)$$

$$\lambda_3 = m - \sqrt{p}\,(\cos\phi - \sqrt{3}\sin\phi)$$

where

$$\phi = \frac{1}{3}\tan^{-1}\frac{\sqrt{p^3 - q^2}}{q} \qquad 0 \le \phi \le \frac{\pi}{3}$$

Compute the eigenvectors. Let λ represent one of the three eigenvalues λ_1, λ_2, λ_3.

$$\vec{C}_1 = \begin{pmatrix} a_{11} - \lambda \\ a_{21} \\ a_{31} \end{pmatrix} \times \begin{pmatrix} a_{21} \\ a_{22} - \lambda \\ a_{32} \end{pmatrix}$$

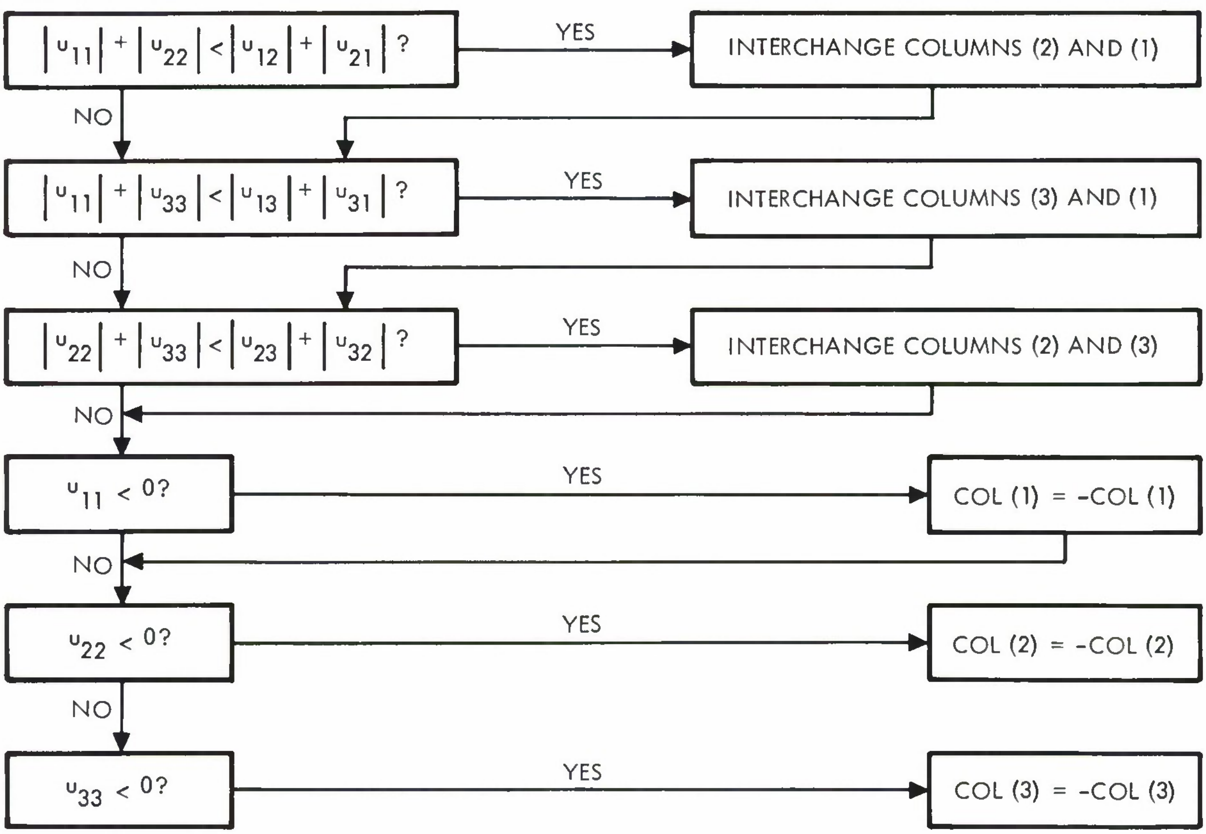
 PRAXIS

$$\vec{C}_2 = \begin{pmatrix} a_{21} \\ a_{22} - \lambda \\ a_{32} \end{pmatrix} \times \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} - \lambda \end{pmatrix}$$

$$\vec{C}_3 = \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} - \lambda \end{pmatrix} \times \begin{pmatrix} a_{11} - \lambda \\ a_{21} \\ a_{31} \end{pmatrix}$$

If $\vec{C}_1 \cdot \vec{C}_2 < 0$; set $\vec{C}_2 = -\vec{C}_2$. If $\vec{C}_1 \cdot \vec{C}_3 < 0$; set $\vec{C}_3 = -\vec{C}_3$. $\vec{u} = 1/3(\vec{C}_1 + \vec{C}_2 + \vec{C}_3)$. $\vec{u} = \vec{u}/|\vec{u}|$ is the eigenvector corresponding to λ.

Letting the three eigenvectors form the columns of the matrix U, the following diagram shows the logic used to maximize the sum of the diagonal elements of U.



Finally, compute ϕ_1, ϕ_2, ϕ_3

$$\phi_1 = \tan^{-1}\left[-\frac{u_{32}}{u_{33}}\right]$$

$$\phi_2 = \sin^{-1}\left[-u_{31}\right]$$

$$\phi_3 = \tan^{-1}\left[-\frac{u_{21}}{u_{11}}\right]$$

SUBROUTINE IDENTIFICATION

A. Title

PRELIM

B. Segment

NRTPOD

C. Called by subroutine

RADR

FUNCTION

The function is to calculate preliminary quantities for the formulation of residuals and partial derivatives of observation with respect to solution parameters.

USAGE

A. Calling sequence

Call PRELIM

B. Input

1. COMMON

a.	PSTAT(4)	Cos ϕ^*
	PSTAT(5)	Sin ϕ^*
	PSTAT(6)	$\alpha g_o + \lambda$ (rad)
	PSTAT(7)	$w_1{}^s$ (e.r.)
	PSTAT(8)	$w_3{}^s$ (e.r.)
b.	PUBS(1)	T (min)
	PUBS(6)	$\dot{R}$ (e.r./min)
c.	TRAJX(1)	x
	TRAJX(2)	y
	TRAJX(3)	z
	TRAJX(4)	$\dot{x}$
	TRAJX(5)	$\dot{y}$
	TRAJX(6)	$\dot{z}$
	TRAJX(7)	$\ddot{x}$
	TRAJX(8)	$\ddot{y}$
	TRAJX(9)	$\ddot{z}$
		TRAJAX (>9) partials of TRAJX (1-9) with respect to P_i, i = 1, NDPR
		P = parameters in the solution vector

d. NDPR — Number of all differential plus initial parameters to solve for (Category 1) including drag parameters ($C_D A/m$'s).

e. TEMP — Temporary storage

f. CWE — Earth's rotational rate

2. Calling sequence

—

C. Output

1. COMMON

PCMR	R = computed slant range
PCSA	Cos A_o
PCSALF	Cos (α_g)
PCSE	Cos E_o
PRSUB1	$R_1 = V_R$
PSNA	Sin A_o
PSNALF	Sin (α_g)
PSNE	Sin E_o
PUDTI	Vector $(\dot{u}_1, \dot{u}_2, \dot{u}_3)$
PUI	Vector (u_1, u_2, u_3)
PV	$\sqrt{v_1^2 + v_2^2}$
PVI	Vector (v_1, v_2, v_3)
PWDTI	Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$
PWDTPP	Partial derivatives
PWI	Vector (w_1, w_2, w_3)
PWPP	Partial derivatives
PR2DØT	$\ddot{R}$
PWDT2P	$\partial\ddot{\bar{W}}/\partial P_i$
PUDDTI	$\ddot{\bar{U}}$
PWDT2	$\ddot{\bar{W}}$

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

COS
SIN
SQRT

B. Program

—

EQUATIONS

The computed orbit positions (x, y, z) and station positions (ϕ^*, λ, h) are processed to produce geocentric and topocentric coordinates of the vehicle in an Earth-fixed coordinate system. Right ascensions of the station for times of observations t_i are

$$\alpha_i = (\alpha_{go} + \lambda) + \omega_e (t_i - t_o)$$

Geocentric position and velocity of the vehicle in Earth-fixed coordinates are

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_i = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i & 0 \\ -\sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \end{bmatrix}_i \begin{bmatrix} \cos\alpha_i & \sin\alpha_i & 0 \\ -\sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} + \omega_e y \\ \dot{y} - \omega_e x \\ \dot{z} \end{bmatrix}$$

$$\begin{bmatrix} \ddot{w}_1 \\ \ddot{w}_2 \\ \ddot{w}_3 \end{bmatrix}_i = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i & 0 \\ -\sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{x} + 2\omega_e \dot{y} - \omega_e^2 x \\ \ddot{y} - 2\omega_e \dot{x} - \omega_e^2 y \\ \ddot{z} \end{bmatrix}$$

The station position in meridian coordinates is provided by the pre-processor module where it is computed from geodetic latitude, ϕ^*, and altitude, h, as follows.

$$A_s = \left(\cos^2 \phi^* + b_e^2 \sin^2 \phi^*\right)^{-1/2}$$

$$B_s = \left(\sin^2 \phi^* + \frac{1}{b_e^2} \cos^2 \phi^*\right)^{-1/2}$$

$$w_1^s = (a_e A_s + h) \cos \phi^*$$

$$w_3^s = (b_e B_s + h) \sin \phi^*$$

where a_e and b_e are the equatorial and polar axes of the reference spheroid respectively.

Topocentric coordinates, direction cosines, and related quantities for the vehicle in meridian plane coordinates system are then

$$q_1 = w_1 - w_1^s \qquad \text{(Topocentric position in equatorial coordinate system)}$$

$$q_2 = w_2$$

$$q_3 = w_3 - w_3^s$$

$$R = \sqrt{q_1^2 + q_2^2 + q_3^2}$$

$$\bar{u} = \begin{cases} u_1 = q_1/r \\ u_2 = q_2/r \\ u_3 = q_3/r \end{cases}$$ (Topocentric direction cosines in equatorial system)

$$\dot{\bar{u}} = \begin{cases} \dot{u}_1 = (\dot{w}_1 - Ku_1)/r \\ \dot{u}_2 = (\dot{w}_2 - Ku_2)/r \\ \dot{u}_3 = (\dot{w}_3 - Ku_3)/r \end{cases}$$

$$K = u_1 \dot{w}_1 + u_2 \dot{w}_2 + u_3 \dot{w}_3$$

$$\ddot{\bar{u}} = \begin{cases} \ddot{u}_1 = \ddot{w}_1 - 2\dot{u}_1 \dot{r} - u_1 \ddot{r} \\ \ddot{u}_2 = \ddot{w}_2 - 2\dot{u}_2 \dot{r} - u_2 \ddot{r} \\ \ddot{u}_3 = \ddot{w}_3 - 2\dot{u}_3 \dot{r} - u_3 \ddot{r} \end{cases}$$

$$\bar{v} = \begin{cases} v_1 = u_2 \\ v_2 = -u_1 \sin \phi^* + u_3 \cos \phi^* \\ v_3 = u_1 \cos \phi^* + u_3 \sin \phi^* \end{cases}$$ (Topocentric direction cosines in horizon system)

$$V = \sqrt{v_1^2 + v_2^2}$$

$$R_1 = VR$$

$$\sin E = v_3$$

$$\cos E = V$$

$$\cos A = v_2/V$$

$$\sin A = v_1/V$$

$$\begin{bmatrix} \frac{\partial w_1}{\partial p_i} \\ \frac{\partial w_2}{\partial p_i} \\ \frac{\partial w_3}{\partial p_i} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial p_i} \\ \frac{\partial y}{\partial p_i} \\ \frac{\partial z}{\partial p_i} \end{bmatrix}$$

If range rate observations are used (PUBS $\neq$ 0), then variational equations in velocity are rotated as follows:

$$\begin{bmatrix} \frac{\partial \dot{w}_1}{\partial p_i} \\ \frac{\partial \dot{w}_2}{\partial p_i} \\ \frac{\partial \dot{w}_3}{\partial p_i} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial \dot{x}}{\partial p_i} + \omega_e \frac{\partial y}{\partial p_i} \\ \frac{\partial \dot{y}}{\partial p_i} - \omega_e \frac{\partial x}{\partial p_i} \\ \frac{\partial \dot{z}}{\partial p_i} \end{bmatrix}$$

where the parameters p_i are the ADBARV conditions at epoch (α_o, δ_o, β_o, A_o, r_o, v_o), and drag parameters ($C_D A/m$).

If range acceleration observations are used, the variational equations are rotated as follows:

$$\begin{bmatrix} \frac{\partial \ddot{w}_1}{\partial p_2} \\ \frac{\partial \ddot{w}_2}{\partial p_i} \\ \frac{\partial \ddot{w}_3}{\partial p_i} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial \ddot{x}}{\partial p_i} + 2\omega_e \frac{\partial \dot{y}}{\partial p_i} - \omega_e^2 \frac{\partial x}{\partial p_i} \\ \frac{\partial \ddot{y}}{\partial p_i} - 2\omega_e \frac{\partial \dot{x}}{\partial p_i} - \omega_e^2 \frac{\partial y}{\partial p_i} \\ \frac{\partial \ddot{z}}{\partial p_i} \end{bmatrix}$$

SUBROUTINE IDENTIFICATION

A. Title

PRTATA

B. Segment

NRTPOD

C. Called by subroutine

APPLY

FUNCTION

The functions are to move the de-augmented $A^T A$ and store by rows as a lower triangular matrix in VSTR(NRTMP), to scale as an input $A^T A$ inverse, and to print the $A^T A$ by MATPT.

USAGE

A. Calling sequence

Call PRTATA

B. Input

1. COMMON

NPR	Total number of parameters to solve for
NATA	Starting location of where the triangular $A^T A$ is stored
NRTMP	Starting location of temporary storage for special handling of the R matrix
NSCALE	Starting location of list of conversion factors which convert from machine to output units and vice-versa
KOUT	Symbolic output tape
MPR	Number of parameters in solution vector (constrained system)

2. Calling sequence

—

C. Output

1. COMMON

VSTR (NRTMP)	Contains the scaled $A^T A$ normal matrix which is output off-line

2. Calling sequence

—

D. Error/action messages

SUBROUTINES USED

A. Library

—

B. Program

HUMAH

MATPT

SUBROUTINE IDENTIFICATION

A. Title

PRUDRV

B. Segment

NRTPOD

C. Called by Subroutine

TRJPRO

FUNCTION

Function is to control the post-processing capability of NRTPOD. The trajectory propagation and covariance matrix update is performed in this post-processing link.

USAGE

A. Calling sequence

Call PRUDRV

B. Input

1. COMMON

ITRJTP	Trajectory Tape
PSTFLG	Columns 51-60 on JDC card
TEMP	Array of temporary storage
TRAJX	Array containing position, velocity accelerations and partials of position, velocity, and acceleration with respect to ADBARV and the two drag layers of current influence
CDAD2M	$C_D A/2m$ — (ft^2/slug)
TRHOA	Density of air at TALT (slugs/ft^3)

2. Calling sequence

—

C. Output

1. COMMON

TG	Time to integrate to (min)
TCRASH	Impact flag

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

TPRNT	Routine to print trajectory block
PRAUPD	Prints and updates covariance and correlation matrices

SUBROUTINE IDENTIFICATION

A. Title

PUPB

B. Segment

NRTPOD

C. Called by subroutine

RADR

FUNCTION

The function of this subroutine is to evaluate the partials of observations with respect to biases of time, sensor latitude, sensor longitude, and sensor altitude. The observation type and the bias type are given in the calling sequence.

USAGE

A. Calling sequence

PUPB (I, J)

B. Input

1. COMMON

COUNT	Number of lines
PCMR	R = computed slant range
PCSA	Cos A_c
PCSALF	Cos (α_g)
PCSE	Cos E_c
PRSUB1	R_1 = VR
PSNA	Sin A_c
PSNALF	Sin (α_g)
PSNE	Sin E_c
PSTAT	Working storage for sensor information
PUDTI	Vector $(\dot{u}_1, \dot{u}_2, \dot{u}_3)$
PUI	Vector (u_1, u_2, u_3)
PUBS	Current observation buffer; ID, time, R or $\ddot{R}$, A, E, $\dot{R}$, type [If type (PUBS(7)] = 1, observation is $\ddot{R}$)

PV	$\sqrt{v_1^2 + v_2^2}$
PVI	Vector (v_1, v_2, v_3)
PWDTI	Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$
PWI	Vector (w_1, w_2, w_3)
PUDDTI	Vector $(\ddot{u}_1, \ddot{u}_2, \ddot{u}_3)$
PWDT2	Vector $(\ddot{w}_1, \ddot{w}_2, \ddot{w}_3)$
TRAJX	x, y, z, x, y, z ....
CWE	Earth's rotational rate (rad/min)
KOUT	Output tape number

2. Calling sequence

$J = 1$ for R or $\ddot{R}$

$= 2$ for A

$= 3$ for E

$= 4$ for $\dot{R}$

$I = 7$ for t_b

$= 8$ for ϕ_b^*

$= 9$ for ℓ_b

$= 10$ for h_b

C. Output

1. COMMON

—

2. Calling sequence

$$\text{A register} = \frac{\partial(\text{variable J})}{\partial(\text{variable I})}$$

D. Error/action messages

PARTIAL ( ) WITH RESPECT ( ) ASKED FOR

Given off-line when I and J exceed current program limits (I = 10, J = 6)

SUBROUTINES USED

A. Library

—

B. Program

LINES Line counter

EQUATIONS

Range (J = 1 and type = 0)

$$\frac{\partial R}{\partial \phi^*} = u_1 w_3^s - u_3 w_1^s \text{ (type 8 bias)}$$

$$\frac{\partial R}{\partial \lambda} = u_1 w_2 - u_2 w_1 \text{ (type 9 bias)}$$

$$\frac{\partial R}{\partial h} = -u_1 \cos \phi^* - u_3 \sin \phi^* \text{ (type 10 bias)}$$

$$\frac{\partial R}{\partial t} = u_1 \dot{w}_1 + u_2 \dot{w}_2 + u_3 \dot{w}_3 \text{ (type 7 bias)}$$

Azimuth (J = 2)

$$\frac{\partial A}{\partial \phi^*} = \frac{\sin A}{R_1} (w_1 \cos \phi^* + w_3 \sin \phi^*) \text{ (type 8 bias)}$$

$$\frac{\partial A}{\partial \lambda} = \frac{-w_1 \cos A + w_2 \sin \phi^* \sin A}{R_1} \text{ (type 9 bias)}$$

$$\frac{\partial A}{\partial h} = 0 \text{ (type 10 bias)} \quad R_1 = VR$$

$$\frac{\partial A}{\partial t} = \frac{1}{V^2} (v_2 \dot{v}_1 - v_1 \dot{v}_2) \text{ (type 7 bias)}$$

Elevation (J = 3)

$$\frac{\partial E}{\partial \phi^*} = \frac{1}{R_1} \left(w_3 \cos \phi^* - w_1 \sin \phi^* - \frac{\partial R}{\partial \phi^*} \sin E \right) \text{ (type 8 bias)}$$

$$\frac{\partial E}{\partial \lambda} = \frac{1}{R_1} \left(w_2 \cos \phi^* - \frac{\partial R}{\partial \lambda} \sin E \right) \text{ (type 9 bias)}$$

$$\frac{\partial E}{\partial h} = -\frac{1}{R_1}\left(1 + \frac{\partial R}{\partial h} \sin E\right) \text{(type 10 bias)}$$

$$\frac{\partial E}{\partial t} = \frac{\dot{u}_1 \cos \phi^* + \dot{u}_3 \sin \phi^*}{\cos E} \quad \text{(type 7 bias)}$$

Range Rate (J = 4)

$$\frac{\partial \dot{R}}{\partial \phi^*} = w_3{}^s \dot{u}_1 - w_1{}^s \dot{u}_3 \quad \text{(type 8 bias)}$$

$$\frac{\partial \dot{R}}{\partial \lambda} = (w_2 \dot{u}_1 - w_1 \dot{u}_2) + (\dot{w}_2 u_1 - \dot{w}_1 u_2) \quad \text{(type 9 bias)}$$

$$\frac{\partial \dot{R}}{\partial h} = -\dot{u}_1 \cos \phi^* - \dot{u}_3 \sin \phi^* \quad \text{(type 10 bias)}$$

$$\frac{\partial \dot{R}}{\partial t} = \ddot{R} = \bar{\dot{u}} \cdot \dot{W} + \bar{u} \cdot \bar{\ddot{W}} \quad \textbf{(type 10 bias)}$$

where

$$\bar{\ddot{w}} = \begin{cases} \ddot{w}_1 = -\omega_e^2 w_1 + 2\omega_e(-\dot{x} \sin \alpha + \dot{y} \cos \alpha) + (\ddot{x} \cos \alpha + \ddot{y} \sin \alpha) \\ \ddot{w}_2 = -\omega_e^2 w_2 + 2\omega_e(-\dot{x} \cos \alpha - \dot{y} \sin \alpha) + (-\ddot{x} \sin \alpha + \ddot{y} \cos \alpha) \\ \ddot{w}_3 = \ddot{z} \end{cases}$$

Range Acceleration [J = 1 and PUBS(7) = 1]

$$\frac{\partial \ddot{R}}{\partial \phi^*} = w_3{}^s \ddot{u}_1 - w_1{}^s \ddot{u}_3 \quad \text{(type 8 bias, I = 8)}$$

$$\frac{\partial \ddot{R}}{\partial \lambda} = (\ddot{u}_1 w_2 - \ddot{u}_2 w_1) + 2(\dot{u}_1 \dot{w}_2 - \dot{u}_2 \dot{w}_1) + (\ddot{w}_2 u_1 - \ddot{w}_1 u_2)$$

(type 9 bias, I = 9)

$$\frac{\partial \ddot{R}}{\partial h} = -\ddot{u}_1 \cos \phi^* - \ddot{u}_3 \sin \phi^* \quad \text{(type 10 bias, I = 10}$$

SUBROUTINE IDENTIFICATION

A. Title

RADR

B. Segment

NRTPOD

C. Called by subroutine

DCITER

FUNCTION

Function is to control region for the formulation of the system of equations to be solved (Ax = B). A is the matrix of partial derivatives of observations with respect to solution variables and B is the vector of observation residuals. RADR also drives those routines which, given A, B, form A^TA, A^TB, and B^TB. It also drives the residuals print routines.

USAGE

A. Calling sequence

Call RADR

B. Input

1. COMMON

IPFRST	0 to indicate first time in RADR
NAROW	Starting location where one row of the augmented matrix (A, B) is stored
NPR	Number of all parameters to solve for
PCMR	Computed slant range
POBCNT	Total number of accepted observations
PRESD	Residuals
PSIG	Sigma list
PUBS	Sensor number, time, R, A, E, $\dot{R}$, $\ddot{R}$, table
PUI	Vector (u_1, u_2, u_3)
NDPR	Number of CAT 1 variables in the solution vector
NPBIS	Pointer to starting location in VSTR of biases being solved for

NPRCD	Starting location of code words denoting biases to be solved for
NSSTB	Starting location of the mean and SOS table of residuals by type and station
MPR	Number of parameters in solution vector (constrained vector)
PDELFG	Array containing editing symbols (*, N, K, S or blank)
PSTAT	Current working sensor array
PR2DOT	Computed range acceleration
PVI	Vector (v_1, v_2, v_3)
PWDTI	Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$
TSUS	Current total SOS
VSTR	Floating point variable storage
CPI	π
C2PI	2π
PCSE	Cos E_c

2. Calling sequence

—

C. Output

1. COMMON

The array VSTR (NATA) contains the total A^TA, A^TB, B^TB.

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

ASIN	Arc sine routine
ATNQ	Arc tangent routine
DRDP	Partials of observations w. r. t. Category 1 variables

LEGS1	Forms A^TA and A^TB given A and B
PIMOD	Principal value of angle between 0 and 2π
PRELIM	Preliminary calculations
PAGE1	Accumulates residuals and prints
LINES	Counts output lines per page
FSIGMA	Functional sigma subroutine
PAROUT	Up, down, cross coordinate system routine
SSTB	Computes mean, SOS of residuals by type and station
NTOM	Linear constraint subroutine
REJECT	Editing of residuals processor
PUPB	Computes partials of observations w.r.t. biases of time, sensor latitude, longitude, and altitude

EQUATIONS

Computation of Observables from Fitted Orbit

The fitted orbit is used to produce computed "observables" for comparison with observations.

$$R = \sqrt{q_1^2 + q_2^2 + q_3^2} \qquad \text{(range)}$$

$$A = \tan^{-1} v_1/v_2 \qquad \text{(azimuth)}$$

$$E = \sin^{-1} v_3 = \cos^{-1} V \qquad \text{(elevation)}$$

$$\dot{R} = \bar{u} \cdot \dot{\bar{W}} \qquad \text{(range rate)}$$

$$\ddot{R} = \dot{\bar{u}} \cdot \dot{\bar{W}} + \bar{u} \cdot \ddot{\bar{W}} \qquad \text{(range acceleration)}$$

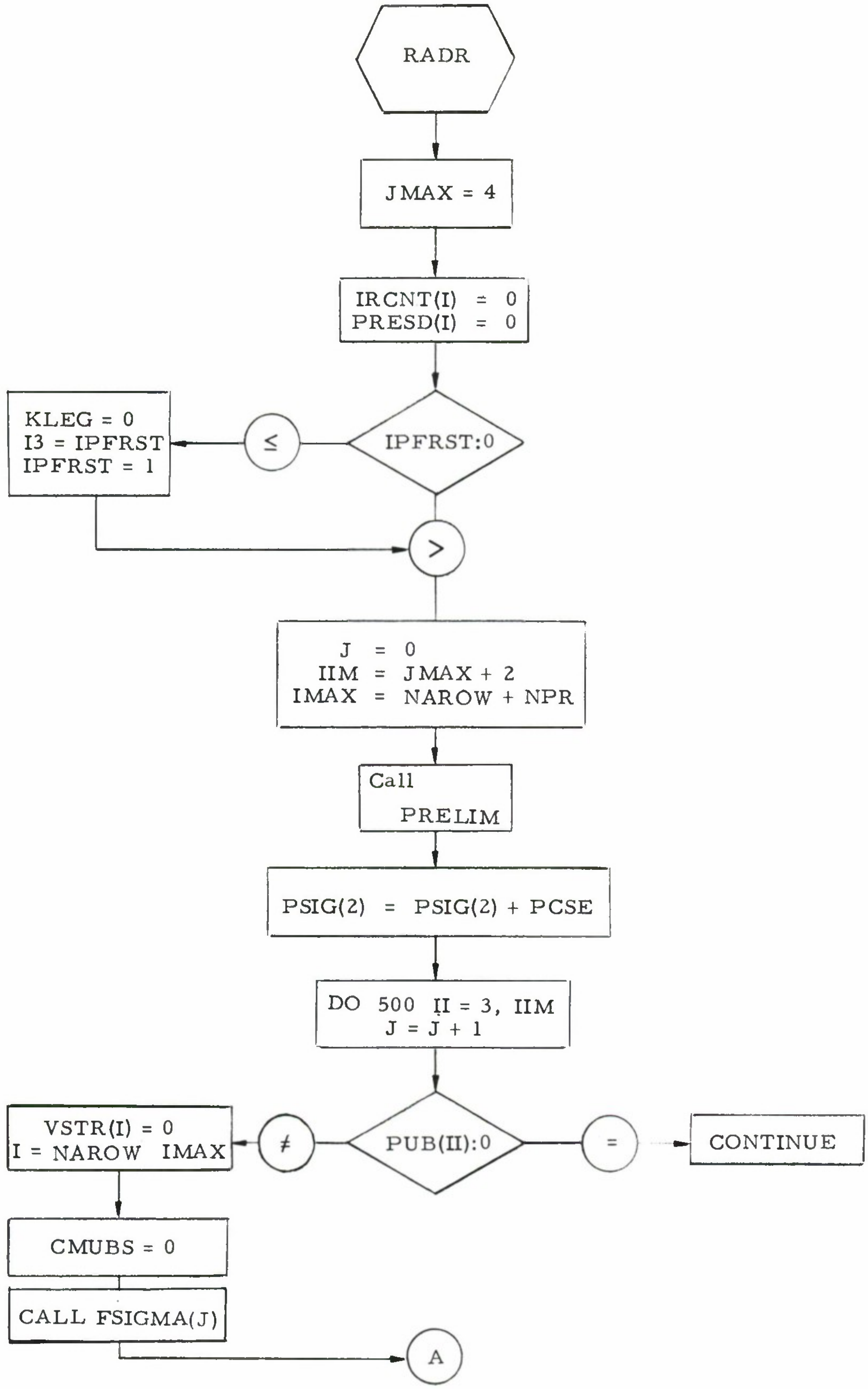


Figure 5-12. RADR Flow Diagram

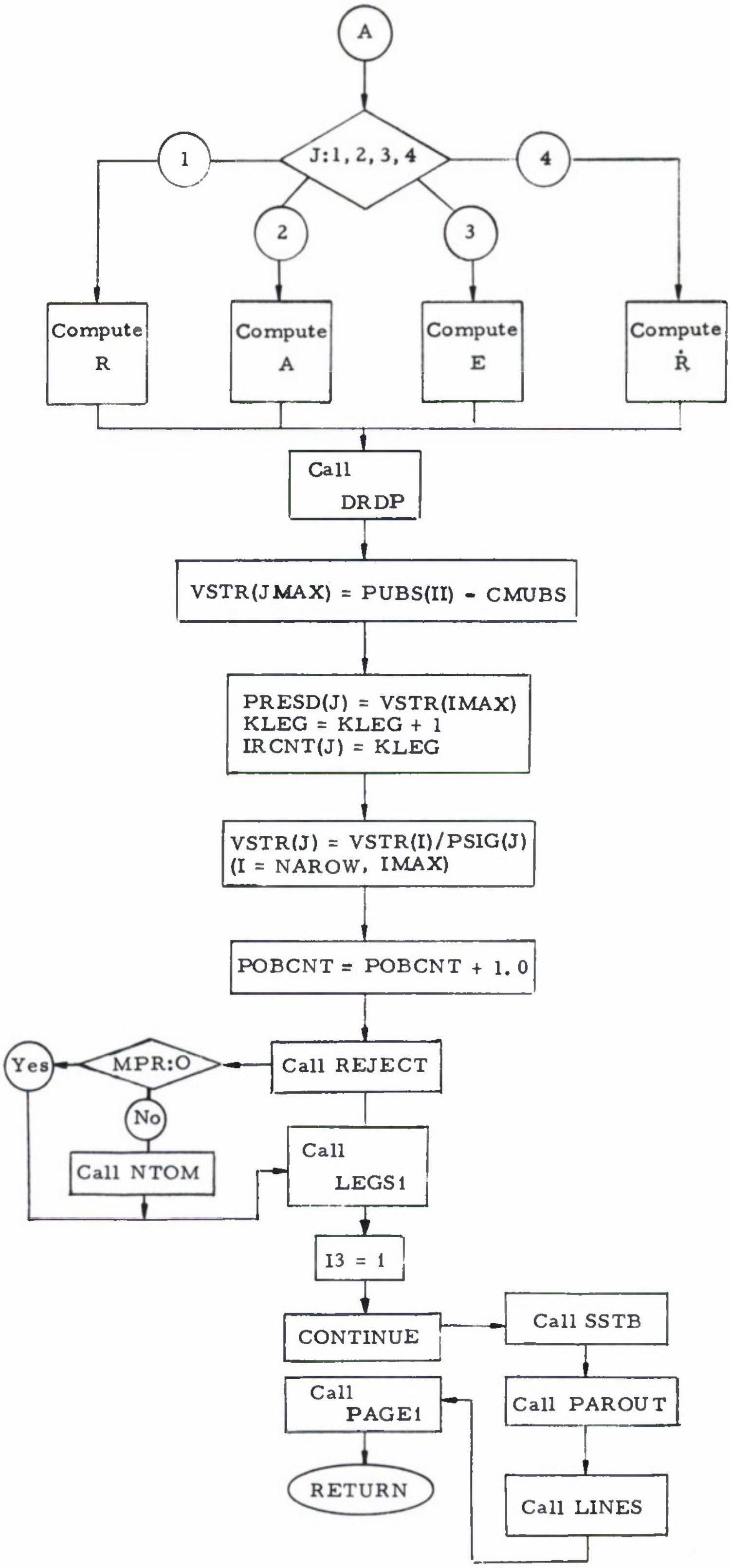


Figure 5-12. RADR Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title

RDDATA

B. Program

NRTPOD - Input processor

C. Called by subroutine

INPUT

FUNCTION

To read off-line into core storage all NAMELIST input and the lunar-solar ephemeris data

USAGE

A. Calling sequence

CALL RDDATA

B. Input

1. Blank COMMON

KIN	Symbolic input tape
KOUT	Symbolic output tape (print)
COM	Variables in BLK1 blank COMMON (See NAMELIST input and Layout of COMMON Storage sections)
DTMAX	A provision for editing residuals by input (See NAMELIST input section)
NDAYS	Number of days of lunar-solar ephemeris input (See NAMELIST input section)
CNSIG	N for N (σ) deletion, a provision for editing residuals by input. (NAMELIST Input)
TIME	A 6-cell array containing epoch time in year, month, day, hour, minutes, seconds (NAMELIST Input)
DELTT	Sets of Δt (See NAMELIST Input section)

NITER	Number of iterations desired in curve fit, nominally = 1 (NAMELIST Input)
TYPE	Indicates type of initial conditions (position and velocity) input to NRTPOD (NAMELIST Input)
BFLAGS	Flags indicating whether the sun and moon are to be included in the trajectory simulation (See NAMELIST Input)
CKRMS	A provision for editing residuals by input (See NAMELIST Input section)

2. Labeled COMMON

/INPP/	
DRAG	$C_D A/m \left(\frac{\text{meter}^2}{\text{kilogram}}\right)$ (NAMELIST Input)
DRAGCD	C_D Coefficient of drag in DRAG (NAMELIST Input)
DRAGA	A - area in DRAG term (meters^2) (NAMELIST Input)
DRAGM	m - mass in DRAG term (kilogram) (NAMELIST Input)
STVEC	Array identifying the initial position and velocity. (See NAMELIST Input section)
CAT1	The CAT1 array indicates to the program the Category 1 variables to be solved for (See NAMELIST Input section)
CAT2	The CAT2 array indicates to the program the Category 2 variables to be solved for (See NAMELIST Input section)
BISES	Bias estimates: (See NAMELIST Input section)
SMAT	A priori normal matrix (See NAMELIST Input)
DELET	Input provided to edit residuals (See NAMELIST Input)
BNDS	Bounds specified to control convergence for each CAT1 or CAT2 variable selected for solution (NAMELIST Input)
ZONAL	Array of flags for callouts of the coefficients of the zonal harmonics - $J_2, \ldots, J_{12}$ (NAMELIST Input)
SECT	Array of flags for callouts of the sectorial harmonics (See NAMELIST Input Section)

TESS	Array of code words for selection of tesseral harmonics (See NAMELIST Input section)
RADPR	Radiation pressure parameter, $\frac{\gamma A}{m}$ $\left(\frac{\text{meter}^2}{\text{kilogram}}\right)$ (See NAMELIST Input)
RPGAM	Radiation pressure parameter, γ, reflectivity constant (NAMELIST Input)
RPA	Radiation pressure parameter, A, effective area of vehicle in square meters (NAMELIST Input)
RPM	Radiation pressure parameter, m, mass of the vehicle in kilograms (NAMELIST Input)
CJ	Zonal harmonics, $J_2, \ldots, J_{12}$. May be altered on input (NAMELIST Input)
CJNM	Coefficients of the sectorial and tesseral harmonics and their associated angles (See NAMELIST Input)
CLAMNN	Array containing values of the angles associated with the coefficients of the tesseral harmonics; $\lambda_2^2, \lambda_3^3, \ldots, \lambda_6^6$ (See NAMELIST Input section)
UPMAT	A priori covariance matrix (See NAMELIST Input)
TPOS	A 60-cell vector containing the position of the moon and sun for NDAYS days TPOS array order is $x_{a_1}, y_{a_1}, z_{a_1}, x_{\odot_1}, y_{\odot_1}, z_{\odot_1}, \ldots$ * $\ldots, x_{a_{NDAYS}}, y_{a_{NDAYS}}, z_{a_{NDAYS}},$ $x_{\odot_{NDAYS}}, y_{\odot_{NDAYS}}, z_{\odot_{NDAYS}}$ Units of earth radii - mean of 1950
TDEL2	A 60-cell vector containing the second central differences of the position ephemeris of the moon and sun for NDAYS days TDEL2 array order is $\delta^2 x_{a_1}, \delta^2 y_{a_1}, \delta^2 z_{a_1}, \delta^2 x_{\odot_1}, \delta^2 y_{\odot_1}, \ldots$

* a - moon
$\odot$ - sun

	$\ldots, \delta^2 x_{\odot NDAYS}, \delta^2 y_{\odot NDAYS},$
	$\delta^2 z_{\odot NDAYS}$
	Units of earth radii - mean of 1950.
TDEL4	A 60-cell vector containing the fourth central differences of the position ephemeris of the moon and sun for NDAYS days TDEL4 array order is
	$\delta^4 x_{a_1}, \delta^4 y_{a_1}, \delta^4 z_{a_1}, \delta^4 x_{\odot_1}, \delta^4 y_{\odot_1},$
	$\delta^4 z_{\odot_1}, \ldots, \delta^4 x_{\odot NDAYS}, \delta^4 y_{\odot NDAYS},$
	$\delta^4 z_{\odot NDAYS}$
	Units of earth radii—mean of 1950.
/EPHCOM/	
XJD	A 10-cell vector containing NDAYS Julian dates. Each Julian date is input mod 2,430,000.0. XJD array order is
	$JD_1, JD_2, JD_3, \ldots, JD_{NDAYS}$
BIJ	Non-zero element of constraint matrix
XIJ	Subscripts for non-zero elements of constraint matrix, B_{ij}.
CI	Additive constants for constraint problem
ALTS	Altitude table for multiple drag (kilometers).
CLAMDA	$C_D A/m$ table corresponding to ALTS table (m^2/kg).
CATLM	Array which flags the drag solution variables.
CHEPS	Tolerence criterion of altitude cut-offs (earth radii).
ECRIT	Minimum elevation to allow steering ephemeris print.

DAYINT	Integer portion of Julian date on mean elements card.
DAYFRC	Fractional portion of Julian date on mean elements card.
RDFLG	Flag which changes output units of $\dot{R}$ and $\ddot{R}$.
SMELM	21-word vector containing the Smithsonian mean elements and their time derivatives.
TNULL	Time to which input elements are to be updated.

3. Calling sequence

—

C. Output

1. Blank COMMON

—

2. Labeled COMMON

—

3. Calling sequence

—

D. Error/action messages

1. Off-line comments

"NO. OF EPHEMERIS DAYS LESS THAN 4, TURN BODIES OFF"

2. On line comment

—

3. Action

If the number of lunar-solar ephemeris days (NDAYS) is greater than 0 and less than 4, the off-line comment is printed and NDAYS is set equal to 0, which in effect turns off computation of perturbative accelerations due to the moon and sun and/or radiation pressure.

RDDATA

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SELECT

B. Program

NRTPOD

C. Called by Subroutines

TRJGEN

FUNCTION

To select the next output time for the trajectory package. This routine is used to select the next observation time during the curve fit portion of NRTPOD, and the next DELTT time for the print-update option.

USAGE

A. Calling sequence

Call SELECT

B. Input

1. COMMON

TEPOCH	Epoch time, minutes from 0 hours
DELTT	8 sets of Δt, T
KONTRL	= 1 if curve fit in progress, = 2 if trajectory print-update
TLIST	Integration list
NDTCT	Counter for DELTT array to indicate next set to be processed
NLAMS	Number of drag parameters in the solution vector
NPXDLM	Starting location in VSTR of d_i vectors, where

$$d_i = \frac{\partial\left(\ddot{x},\ \ddot{y},\ \ddot{z}\right)_t}{\partial\left(\frac{C_D A}{m}\right)_i} \qquad i = 1, \ldots, NLAM$$

NLAM	Total number of entries in the altitude $C_D A/m$ table

NPALM — Starting location in VSTR of b_i vectors, where

$$b_i = \left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_t}{\partial(\text{ADBARV})_{t_o}}\right]^{-1} \left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_t}{\partial\left(\frac{C_D A}{m}\right)_i}\right]$$

2. Calling sequence

—

C. Output

TG	The time of the next output, minutes from 0 hours day of epoch
PUBS	The next observation (if KONTRL = 1)
TUBSEF	Non-zero if the end of the observation tape has been sensed (if KONTRL = 1)

D. Error/action messages

—

E. Internal Storage

1. COMMON

TMINUS	This flag is used when there are pre-epoch times to be processed. When the first pre-epoch time is encountered this flag is set to 1 and the integration is initialized in the backward time direction. When the first post-epoch time is encountered, re-initialization of the integration at epoch will take place if TMINUS is set to 1. Initially, TMINUS is assumed 0.
NDTCT	Incremented internally

SUBROUTINES USED

A. Library

—

B. Program

UBSGET	Processes observation tape
SETIC	Initializes integration list at epoch

SUBROUTINE IDENTIFICATION

A. Title

SENRD

B. Segment

NRTPOD - Input processor

C. Called by subroutine

LODSEN

FUNCTION

Function is to read the sensor cards (6 types) and to build a temporary buffer (DTMP) for biases and weights by station.

USAGE

A. Calling sequence

CALL SENRD (SEOF)

B. Input

1. Blank COMMON

KIN	Symbolic input tape
KOUT	Symbolic output tape
PREFLG	NRTPOD control flags (col 31 - 40 on JDC)

2. Labeled COMMON

/TEMP/ TEMP	Internal temporary storage

3. Calling sequence

—

C. Output

1. Labeled COMMON

/INPP/	
NDTMP	Counter on DTMP buffer for biases and weights by station.
DTMP (51)	Station ID
(52)	Azimuth bias (deg)
(53)	Elevation bias (deg)
(54)	Range bias (km)
(55)	Range bias (km/sec)
(56)	Range acceleration bias (m/sec^2)

(57)	time bias (sec)
(58)	σ_R standard deviation in range
(59)	σ_A standard deviation in azimuth
(60)	σ_E standard deviation in elevation
(61)	$\sigma_{\dot{R}}$ standard deviation in range rate
(62)	$\sigma_{\ddot{R}}$ standard deviation in range acceleration
(63 ... 74) ... (75 ... 86)	Repeated for each input sensor (Maximum of 25 sensors allowed)

2. Calling sequence

SEOF	Flag indicating whether all sensor cards have been read. = -1. More sensors to be read = +1. END sensor card has been detected. No more sensor cards to be read.

D. Error/action messages

1. Off-line comment

"NO. OF SENSORS GREATER THAN MAX ALLOW. --- IGNORE."

2. On-line comment

—

3. Action

Ignores processing of previous sensor data, and proceeds to the next sensor card.

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SETCON

B. Segment

NRTPOD - Input Processor

C. Called by subroutine

INPUT

FUNCTION

To set up nominal values of program control constants, potential model constants, scale factors, and symbolic tape assignments.

USAGE

A. Calling sequence

CALL SETCON

B. Input

1. COMMON

—

2. Calling sequence

—

C. Output

1. Blank COMMON

CWE	Earth's rotational rate (radians/min)
CELLIP	Ellipticity of the earth
CMU	GM of the earth (e.r.^{3}/min^2)
CGMR	GM ratios (MOON GM/EARTH GM, SUN GM/EARTH GM)
CFTER	ft/e.r.
CKMFT	km/ft
CKMER	km/e.r.
CMTER	meters/e.r.
CDEG	degrees/radian
CFTNM	ft/n mi
CNMER	n mi/earth radii
CDAYMN	12-cell array denoting the number of days in each month
CPI	π
C2PI	2π

KOUT	Output tape number (print)
KIN	Input tape number
MT	Observations tape number
NOUT	Scratch tape not used at present by NRTPOD
ITRJTP	Trajectory ephemeris tape number
CHMAX	Maximum integration step size
CHMIN	Minimum integration step size
CYMIN	Parameter for variable step integration
CER	Parameter for variable step integration
CBE	b_e = 1. - CELLIP
CRASHE, CRASHM	Impact flags used by subroutine TRAJ
CJD50	Julian date Jan 0, 1950
COMLST	Size of variable storage
CFTEPS	RMS convergence criterion
DTMAX	Editing parameter - maximum allowable observation time from epoch (days)
TSTEP	Initial integration step size (min)
BFLAGS	Flags indicating bodies (moon and sun) to be considered
SKIP	If 0, always set FLVE = 0, if non-zero, set FLVE accordingly
CKRMS	A provision for editing residuals by input
CNSIG	N for $N \cdot \sigma$ deletion
NRRR	Ratio of Runge-Kutta step to Cowell step
FLVE	If non-zero, skip VAREQ
CHEPS	Criterion on altitude cut-offs

2. Labeled COMMON

/INPP/

SECT	Array of cells used for callouts of the sectorial harmonics, non-zero to include the desired harmonic
CJ	Values of the coefficients of the zonal harmonics $J_2, \ldots, J_{12}$.
ZONAL	Array of cells used for callouts of the coefficients of the zonal harmonics
CLAMNN	Array containing the angles associated with the coefficients of the tesseral harmonics
CJNM	6 x 6 array containing the coefficients of the sectorial and tesseral harmonics and their associated angles

3. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SETIC

B. Segment

NRTPOD

C. Called by Subroutines

SELECT

TRJGEN

FUNCTION

The function is to initialize the integration list and other parameters which must be re-initialized each time the integration is re-started.

USAGE

A. Calling sequence

Call SETIC

B. Input

1. COMMON

NDPR	Total number of Category 1 variables
TEPOCH	Minutes from midnight day of epoch
TSTEP	Starting step size for the numerical integration in minutes
TICRT	x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ of the vehicle at epoch in earth radii and e.r./min
NLAM	Total number of entries in the altitude C_DA/m table
ALT	Two altitude layers bounding the current region of influence (e.r.)

2. Calling sequence

—

C. Output

1. COMMON

TG	Time to integrate to (min)
TCRASH	Impact flag = 0 not impacted $\neq$ 0 impact

TLIST	Numerical integration working storage
TMINUS	Flag indicating backward integration
PMAT, VMAT	Arrays used in variational equation formulation, initialized at 0
FLVE	Flag for variational equations compatation

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

DAUX
FVE
HEIGHT
HINT
VPERT

SUBROUTINE IDENTIFICATION

A. Title

STEER

B. Segment

NRTPOD

C. Called by subroutine

TPRNT

FUNCTION

Function is to compute radar steering ephemeris for NRTPOD and print summary values.

USAGE

A. Calling sequence

Call STEER

B. Input

1. COMMON

CBE	Semi-minor axis of earth, b_e
CELLIP	Ellipticity of the earth
CKMER	Value of km per earth radii
CKMFT	Conversion from ft to km
CDEG	Conversion from radians to degrees
KOUT	Current output unit
NDPR	Total number of CAT1 variables to solve for
NSTAT	Pointer to first station of master sensor table in VSTR
NMSTAT	Number of entries per station of master sensor table
CDAD2M	Drag parameter
ECRIT	Minimum elevation to allow printing

RDFLG	Flag which changes output units of RDOT, R2DOT (see equations) from (km/min) and (km/min^2) to (km/sec) and (mt/sec^2).
	when RDFLG = 0 The output units of RDOT and R2DOT are (km/min) and (km/min^2) respectively
	RDFLG $\neq$ 0 The output units of RDOT and R2DOT are (km/sec) and (mt/sec^2) respectively
TG	Time to integrate to
TRAJX	Output from TRAJ: x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$, etc.
TRHOA	Density, kg/m^3
PUBS	Observation vector ID, time, R, A, E, $\dot{R}$, type or ID, time, $\ddot{R}$, 0, 0, 0, type
PCMR	R = computed slant range (e. r.)
PSTAT	Working storage for current station in master sensor table
PUI	Vector (u_1, u_2, u_3)
PVI	Vector (v_1, v_2, v_3)
PWDTI	Vector ($\dot{w}_1$, $\dot{w}_2$, $\dot{w}_3$)
PR2DOT	Second time derivative of range

2. Calling sequence

—

C. Output

1. COMMON

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

ASIN	Arcsin
ATNQ	Double argument arctan
SQRT	Square root

B. Program

DOT	Dot product
PIMOD	Angle moded 0 to 2π
PRELIM	Quantities of residuals and partial derivatives of observations
RADSQ	Find the square of the radius

EQUATIONS

$$\text{Altitude} = R - R_e = h$$

$$h = R - \frac{b_e}{\left[1 - \epsilon(2 - \epsilon)\left(\frac{x^2 + y^2}{R^2}\right)\right]^{1/2}}$$

$$R = \sqrt{q_1^2 + q_2^2 + q_3^2} \qquad \text{(range)}$$

$$A = \tan^{-1} v_1/v_2 \qquad \text{(azimuth)}$$

$$E = \sin^{-1} v_3 \qquad \text{(elevation)}$$

$$\dot{R} = \overline{u} \cdot \overline{\dot{W}} \qquad \text{(range rate)}$$

$$\ddot{R} = \overline{\dot{u}} \cdot \overline{\dot{W}} + \overline{u} \cdot \overline{\ddot{W}} \qquad \text{(range acceleration)}$$

The vectors and vector components which are used to calculate the radar observables above are defined in subroutine PRELIM.

Subroutine STEER is called from TPRNT to calculate and print the radar steering ephemeris. The values printed are either obtained through COMMON or are calculated on a call to PRELIM. All the values are converted to the proper output units. The outputs are:

h	Height of vehicle above the ellipsoid (km)
$C_D A/m$	The ballistic coefficient (m^2/kg)
ρ	Atmospheric density (kg/m^3)
R	Slant range to vehicle (km)
AZ	Azimuth of vehicle (degrees)
E	Elevation of vehicle (degrees)
$\dot{R}$	Time rate of change of range (km/min)
$\ddot{R}$	Range acceleration of vehicle (km/min^2)
	Range rate and range acceleration may be output in units of (km/sec) and (mt/sec^2) respectively if RDFLG is set non-zero.

SUBROUTINE IDENTIFICATION

A. Title

TRAJ

B. Program

NRTPOD

C. Called by Subroutine

TRJGEN

FUNCTION

Integrate the equations of motion and up to 24 variational equations to a specified time. The routine uses Runge-Kutta as a starter to build eighth order difference tables for a Cowell method of numerical integration. The routine will automatically exit with a flag set to indicate earth impact.

USAGE

A. Calling sequence

Call TRAJ(TN)

B. Input

1. COMMON

HMAX	Maximum allowable step size
HMIN	Minimum allowable step size
ER, YMIN	Step size test parameters; See method
TLIST	Input and storage, at output values consistent with T
CMU	GM of earth (Earth radii and minutes)
CRASHB	Ellipticity of earth
CRASHE	1×10^{-8}
CRASHM	Altitude below which impact test will be made (earth radii)
NDPR	The number of variational parameters in the integration list
NRRR	Non-zero if fixed step Runge-Kutta desired
SKIP	If 0, evaluate variational equations only on "predictor" steps

2. Calling sequence

TN Time to integrate to (Minutes from epoch)

C. Output

1. COMMON

TRAJX(1-3)	x, y, z Output ... consistent with TN or impact time
TRAJX(4-6)	$\dot{x}, \dot{y}, \dot{z}$
TRAJX(7-9)	$\ddot{x}, \ddot{y}, \ddot{z}$
TRAJX(10-15)	$\delta_1 x, \delta_1 y, \delta_1 z, \delta_1 \dot{x}, \delta_1 \dot{y}, \delta_1 \dot{z}$ first variation
TRAJX(16-21)	$\delta_2 x, \delta_2 y, \delta_2 z, \delta_2 \dot{x}, \delta_2 \dot{y}, \delta_2 \dot{z}$ second variation
. . .	
TRAJX(52-57)	$\delta_8 x, \delta_8 y, \delta_8 z, \delta_8 \dot{x}, \delta_8 \dot{y}, \delta_8 \dot{z}$ eighth variation
TCRASH	Set non-zero if impact occurs
FLVE	Non-zero to indicate predictor steps
PR2DPI	8 x 3 array of partial derivatives of accelerations with respect to ADBARV and the two drag layers of current influence (λ_i and $\lambda_i + 1$)

2. Calling sequence

—

SUBROUTINES USED

A. Program

DAUX

COMMENTS

The integration list must be initialized before calling TRAJ. If impact occurs, the output is at the impact time, not TN. The initialization flag set non-zero externally, is returned zero by TRAJ.

COMMON (TLIST) Storage

TLIST	Program Tag	Description			
1	FLAG	Initialization parameter—initialize when nonzero			
2	T	Current time			
3	H	Current step size			These values must be supplied when FLAG $\neq 0$
4-30	Y(1-27)	$y_1, y_2, \ldots, y_n$			
31-57	YP(1-27)	$\dot{y}_1, \dot{y}_2, \ldots, \dot{y}_n$			
58-84	YPP(1-27)	$\ddot{y}_1, \ddot{y}_2, \ldots, \ddot{y}_n$	DAUX stores 2nd der.		
85-192	TR(1-27, 1-4)	Intermediate storage			N= 3(NDPR + 1)
193-489	DIF	Difference table		During Runge-Kutta phase	
	(1, 1-27)	$\nabla^8 f_i$		$\ddot{y}_{io}$	
	(2, 1-27)	$\nabla^7 f_i$	as I = 1, N	$\ddot{y}_{i1}$	as I = 1, N
	(3, 1-27)	$\nabla^6 f_i$		$\ddot{y}_{i2}$	
	(4, 1-27)	$\nabla^5 f_i$		$\ddot{y}_{i3}$	
	(5, 1-27)	$\nabla^4 f_i$	See method for description of this table.	$\ddot{y}_{i4}$	These values are saved during 8NR Runge-Kutta steps.
	(6, 1-27)	$\nabla^3 f_i$		$\ddot{y}_{i5}$	
	(7, 1-27)	$\nabla^2 f_i$		$\ddot{y}_{i6}$	
	(8, 1-27)	$\nabla^1 f_i$		$\ddot{y}_{i7}$	
	(9, 1-27)	$f_i = y$		$\ddot{y}_{i8}$	
	(10, 1-27)	$'F_i$		y_{i4}	
	(11, 1-27)	$''F_i$		$\dot{y}_{i4}$	

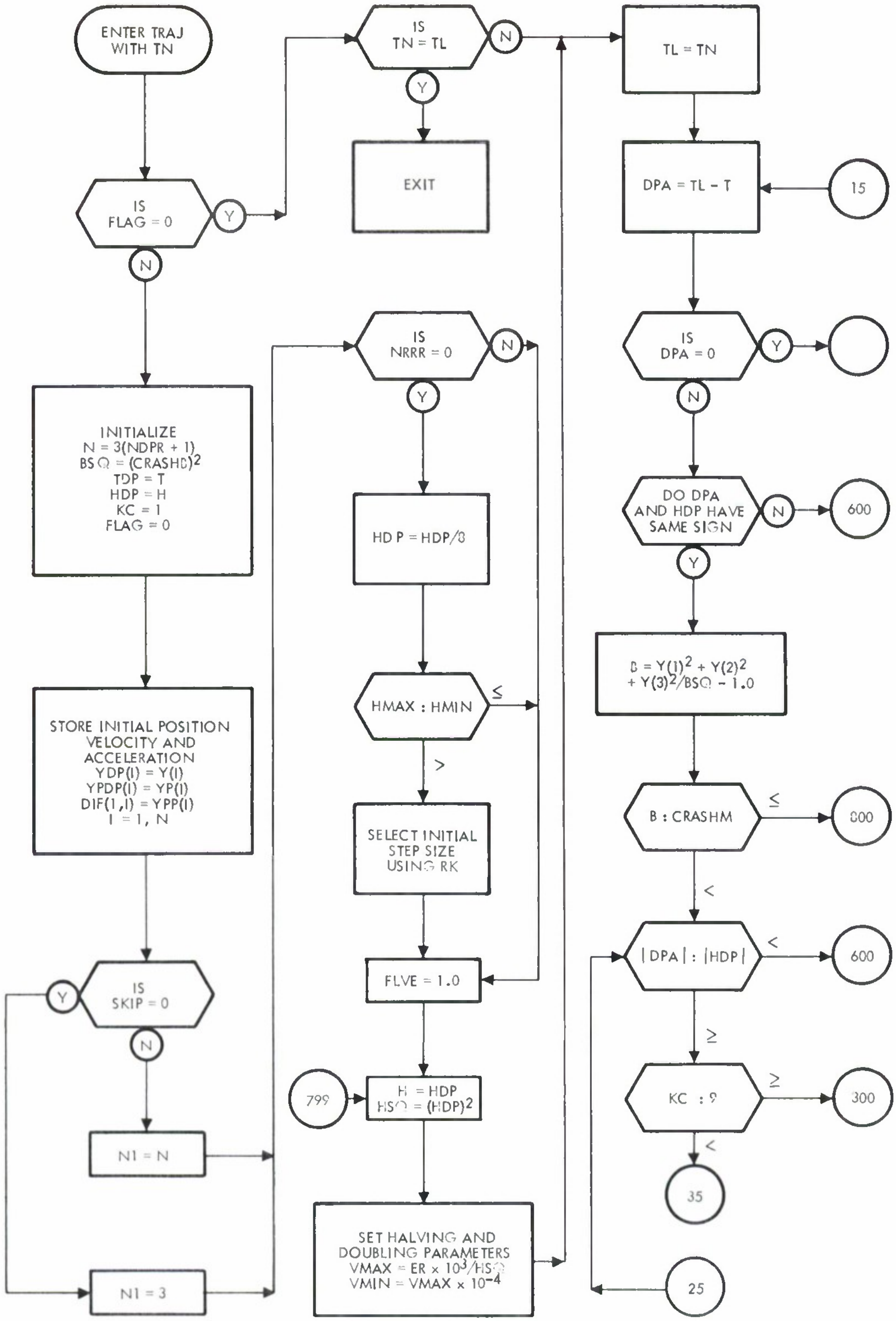


Figure 5-13. TRAJ Flow Diagram

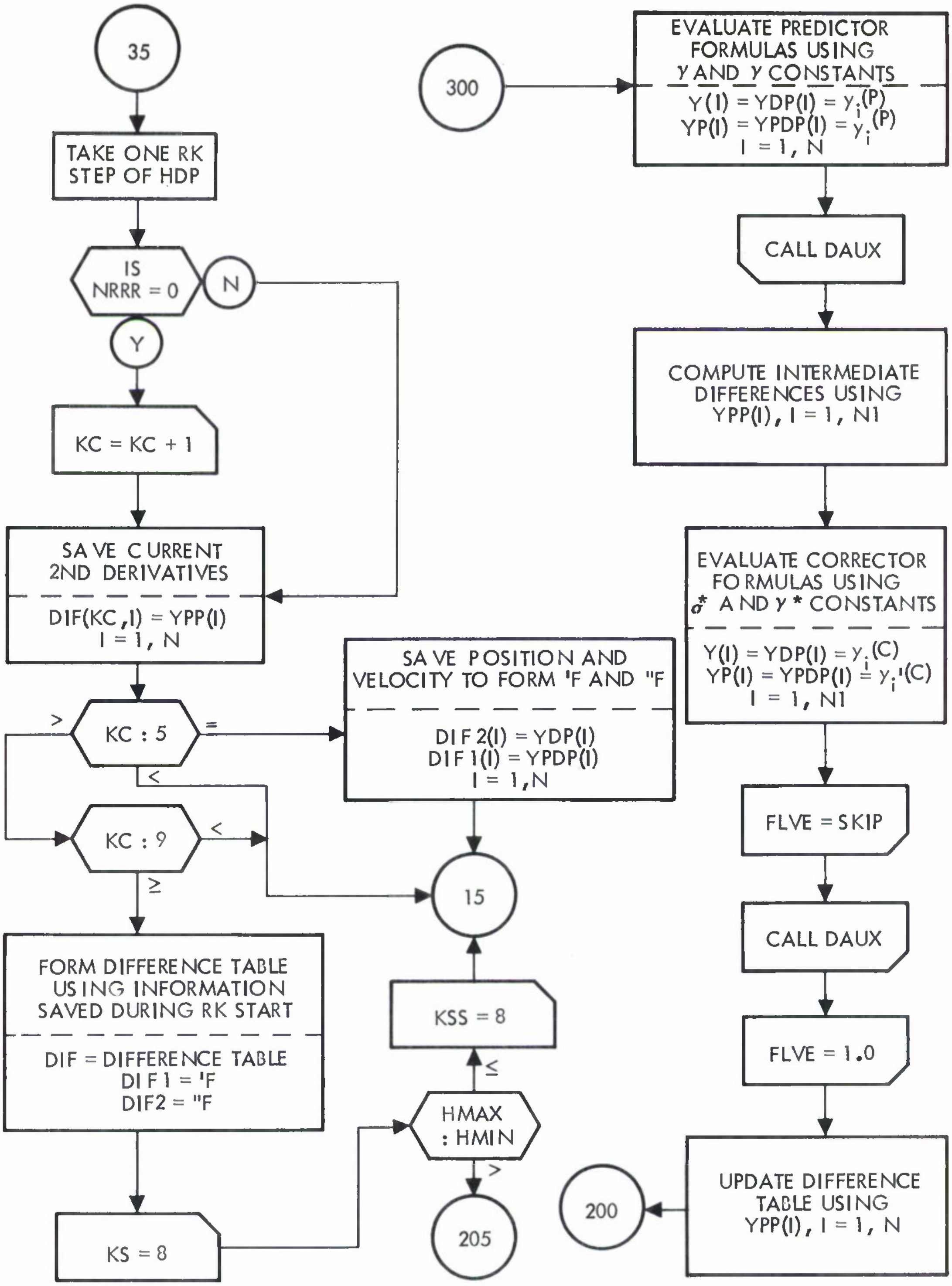


Figure 5-13. TRAJ Flow Diagram (Continued)

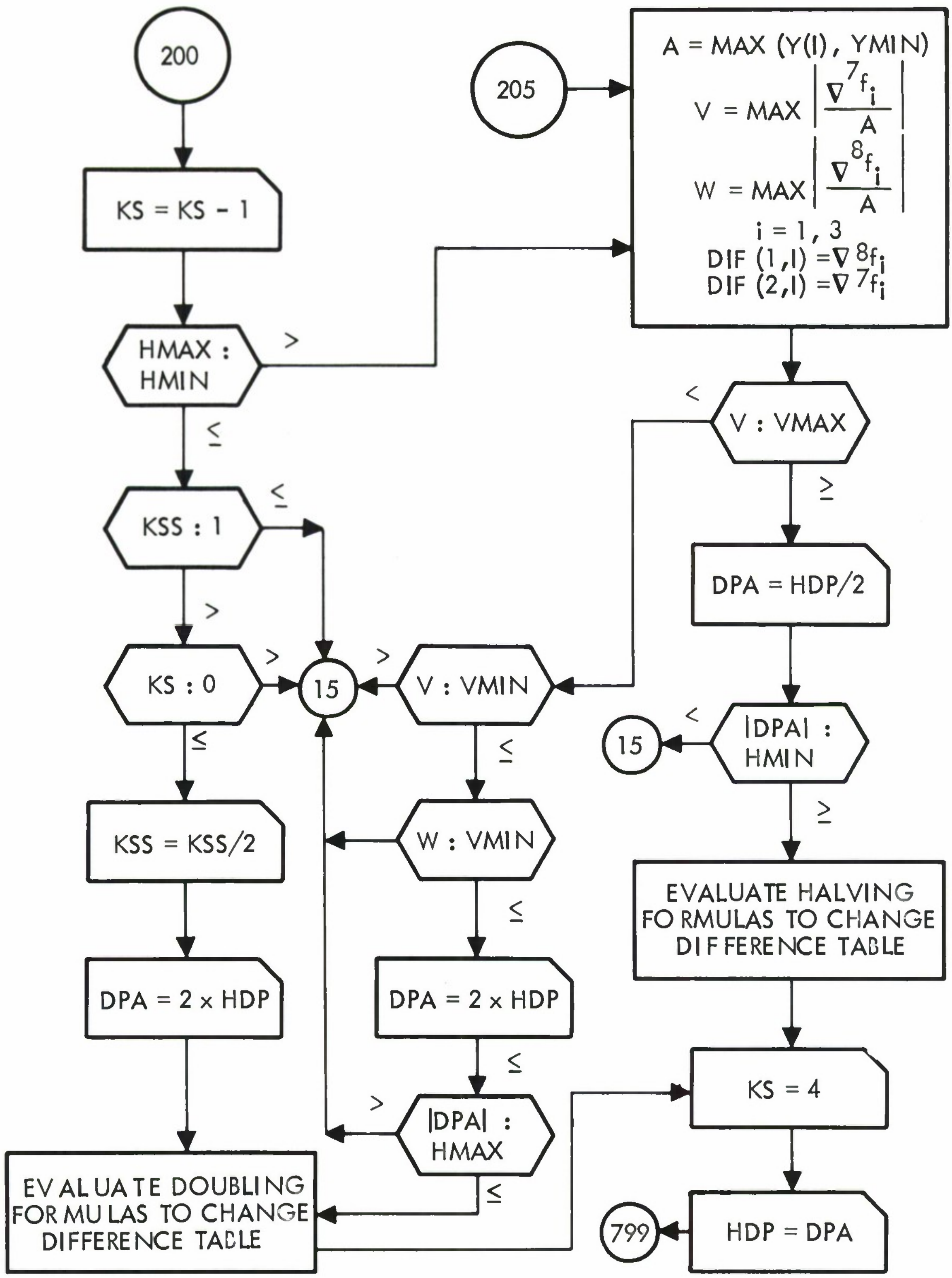


Figure 5-13. TRAJ Flow Diagram (Continued)

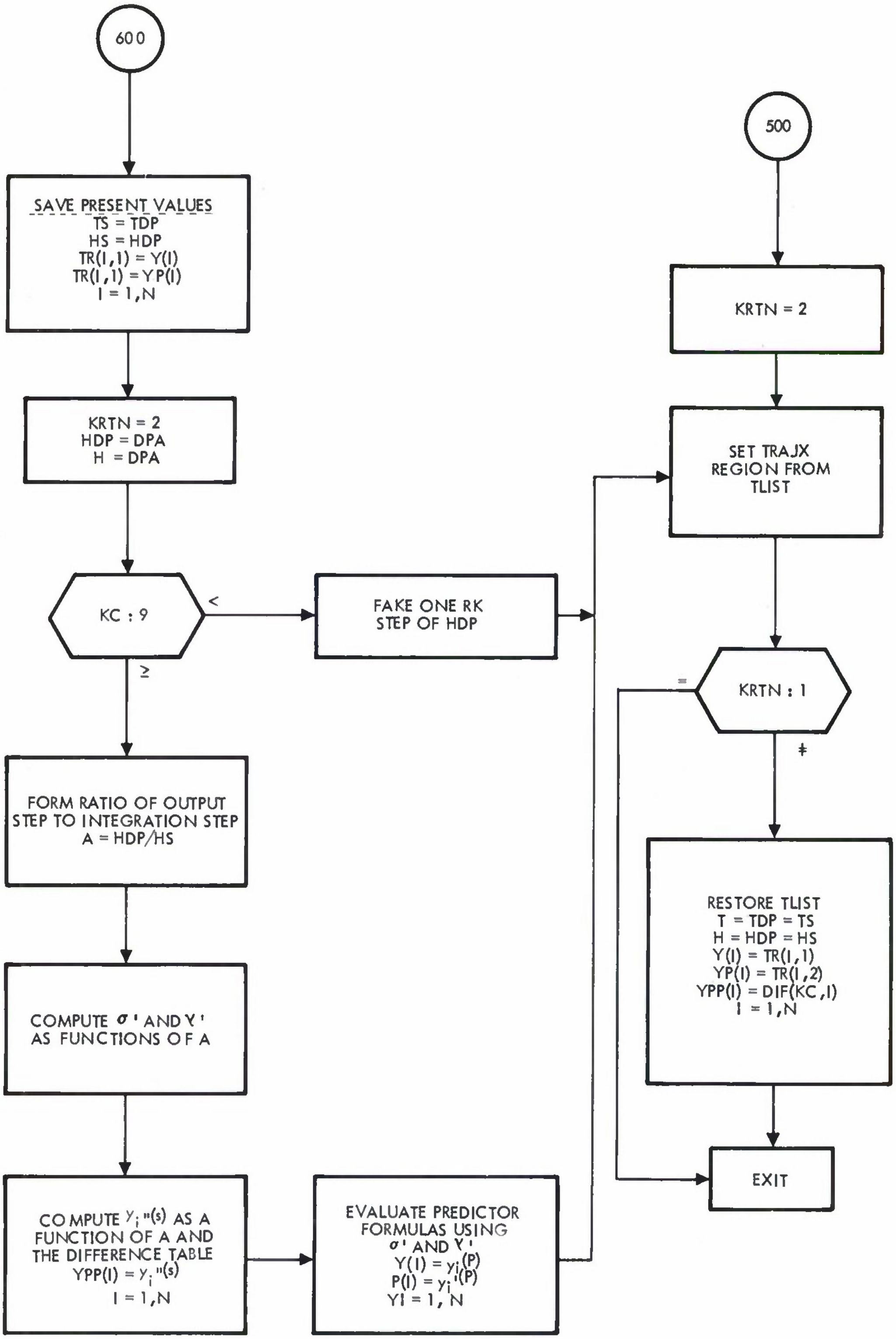


Figure 5-13. TRAJ Flow Diagram (Continued)

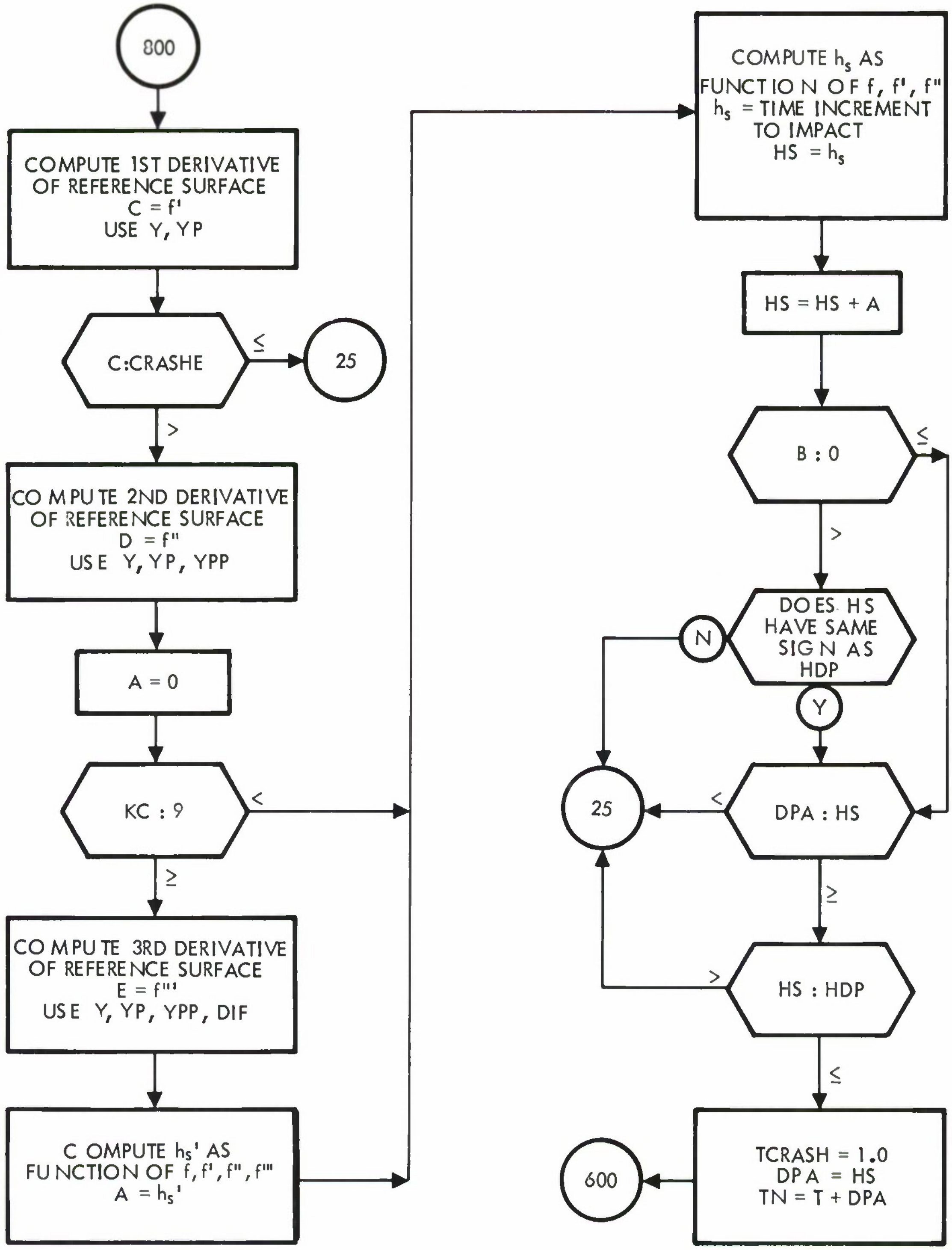


Figure 5-13. TRAJ Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title

TRJGEN

B. Segment

NRTPOD

C. Called by Subroutine

TRJGEN

FUNCTION

Driver for the trajectory link. Controls the logic associated with the trajectory integration and the generation of the trajectory tape.

USAGE

A. Calling sequence

Call TRJGEN

B. Input

1. COMMON

KOUT	Output tape (print)
TEPOCH	Time of epoch, minutes from 0 hours
DTMAX	Maximum allowable time interval for an observation - in days since epoch
TUBSEF	Flag denoting when the last observation has been processed from tape. Set ≠ 0 when "end of file" encountered.
PLSTSN	Station ID for previous observation
TG	Integration time to go. Minutes from 0 hours, day of epoch
TCRASH	Flag indicating earth impact. Non-zero if inpact has occurred
KONTRL	Flag indicating mode of NRTPOD KONTRL = 1 Execute TRJGEN for curve fit and trajectory KONTRL = 2 Execute TRJGEN for trajectory only

2. Calling sequence

—

C. Output

1. COMMON

—

2. Calling sequence

—

D. Error/action messages

1. Action messages

"START TRAJECTORY"

and

"END TRAJECTORY"

Occur when the program begins executing the trajectory link and when execution of the trajectory link terminates

SUBROUTINES USED

A. Library

—

B. Program

SETIC	Initializes integration lists
SELECT	Selects next observation
PARSET	Sets up the PSTAT sensor information array from master sensor table for current observation
TRAJ	Integration subroutine
TRJTAP	Writes trajectory tape
FALSI	Determines altitude cut-offs
PLAMDA	Computes the required partial derivatives of position, velocity, and acceleration with respect to drag
TRJOUT	Prepares a variable length trajectory record to be written on the trajectory tape

SUBROUTINE IDENTIFICATION

A. Title

TRJGET

B. Segment

NRTPOD

C. Called by subroutine

DCITER

FUNCTION

TRJGET reads one trajectory record from the trajectory tape and, if necessary, sets the impact control flag.

USAGE

A. Calling sequence

Call TRJGET (TG)

B. Input

1. COMMON

ITRJTP	Trajectory tape number
TEPOCH	Minutes from midnight day of epoch to epoch

2. Calling sequence

TG	Observation time for which a corresponding trajectory record is to be read

3. Tape input

The trajectory tape generated by the trajectory segment

C. Output

1. COMMON

A(I)	Variable length trajectory array I = 1, . . . , N

where

A (1-9)	$x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}$

(10 - 15) $\frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}, \frac{\partial \dot{x}}{\partial \alpha}, \frac{\partial \dot{y}}{\partial \alpha}, \frac{\partial \dot{z}}{\partial \alpha}$

.
.
.

(40 - 45) $\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}, \frac{\partial \dot{x}}{\partial v}, \frac{\partial \dot{y}}{\partial v}, \frac{\partial \dot{z}}{\partial v}$

$\frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1}, \frac{\partial \dot{x}}{\partial \lambda_1}, \frac{\partial \dot{y}}{\partial \lambda_1}, \frac{\partial \dot{z}}{\partial \lambda_1}$

(46 to 6*NDPR + 9)

.
.
.

$\frac{\partial x}{\partial \lambda_{NLAMS}}, \frac{\partial y}{\partial \lambda_{NLAMS}}, \frac{\partial z}{\partial \lambda_{NLAMS}},$

$\frac{\partial \dot{x}}{\partial \lambda_{NLAMS}}, \frac{\partial \dot{y}}{\partial \lambda_{NLAMS}}, \frac{\partial \dot{z}}{\partial \lambda_{NLAMS}},$

(6* NDPR + 10 to 9*NDPR + 9) $\frac{\partial \ddot{x}}{\partial \alpha}, \frac{\partial \ddot{y}}{\partial \alpha}, \frac{\partial \ddot{z}}{\partial \alpha}$

.
.
.

$\frac{\partial \ddot{x}}{\partial v}, \frac{\partial \ddot{y}}{\partial v}, \frac{\partial \ddot{z}}{\partial v}$

$\frac{\partial \ddot{x}}{\partial \lambda_1}, \frac{\partial \ddot{y}}{\partial \lambda_1}, \frac{\partial \ddot{z}}{\partial \lambda_1}$

.
.
.

$\frac{\partial \ddot{x}}{\partial \lambda_{NLAMS}}, \frac{\partial \ddot{y}}{\partial \lambda_{NLAMS}}, \frac{\partial \ddot{z}}{\partial \lambda_{NLAMS}}$

and NLAMS is the total number of $\frac{C_D A}{m}$'s appearing in the solution vector.

	=	-1, if impact is pre-epoch
TCRASH	=	1, if impact is post-epoch
	=	0, if no impact

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

.FBLT.

.FRDB.

.FVIO.

SUBROUTINE IDENTIFICATION

A. Title

TRJOUT

B. Segment

NRTPOD

C. Called by subroutines

PLAMDA
TRJGEN

FUNCTION

Prepares a variable length trajectory tape record to be written on the trajectory tape at each observation and altitude cut-off point.

USAGE

A. Calling sequence

Call TRJOUT

B. Input

1. COMMON

VSTR	Variable storage
PR2DPI (1-3)	$\frac{\partial \ddot{x}}{\partial \alpha}, \quad \frac{\partial \ddot{y}}{\partial \alpha}, \quad \frac{\partial \ddot{z}}{\partial \alpha}$
(4-6)	$\frac{\partial \ddot{x}}{\partial \delta}, \quad \frac{\partial \ddot{y}}{\partial \delta}, \quad \frac{\partial \ddot{z}}{\partial \delta}$
	$\vdots \qquad \vdots \qquad \vdots$
(16-18)	$\frac{\partial \ddot{x}}{\partial v}, \quad \frac{\partial \ddot{y}}{\partial v}, \quad \frac{\partial \ddot{z}}{\partial v}$
(19-21)	$\frac{\partial \ddot{x}}{\partial \lambda_j}, \quad \frac{\partial \ddot{y}}{\partial \lambda_j}, \quad \frac{\partial \ddot{z}}{\partial \lambda_j}$
(22-24)	$\frac{\partial \ddot{x}}{\partial \lambda_{j+1}}, \quad \frac{\partial \ddot{y}}{\partial \lambda_{j+1}}, \quad \frac{\partial \ddot{z}}{\partial \lambda_{j+1}}$

Where λ_j, λ_{j+1} are the current $(C_D A/m)$ drag parameters in the region of influence

NLAMS		Number of C_DA/m drag parameters in the solution vector
NLID		Pointer to location in variable storage where the identifiers for the (C_DA/m) drag parameters appearing in the solution vector are stored
NPXLM		Pointer to location in variable storage where the $\dfrac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial\left(\dfrac{C_DA}{m}\right)_i}$ are stored $i = 1, \ldots, NLAM$
NLAM		Total number of entries in the altitude C_DA/m table
NPXDLM		Pointer to location in variable storage where the $\dfrac{\partial(\ddot{x}, \ddot{y}, \ddot{z})}{\partial\left(\dfrac{C_DA}{m}\right)_i}$ are stored $i = 1, \ldots, NLAM$
NDPR		Total number of CAT1 variables ($\alpha\,\delta\,\beta$ARV and C_DA/m's) in the solution vector
TRAJX		Integration coordinates referenced to some time, t.
	(1-9)	$x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}$
	(10-15)	$\dfrac{\partial x}{\partial \alpha}, \dfrac{\partial y}{\partial \alpha}, \dfrac{\partial z}{\partial \alpha}, \dfrac{\partial \dot{x}}{\partial \alpha}, \dfrac{\partial \dot{y}}{\partial \alpha}, \dfrac{\partial \dot{z}}{\partial \alpha}$ $\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$
	(40-45)	$\dfrac{\partial x}{\partial v}, \dfrac{\partial y}{\partial v}, \dfrac{\partial z}{\partial v}, \dfrac{\partial \dot{x}}{\partial v}, \dfrac{\partial \dot{y}}{\partial v}, \dfrac{\partial \dot{z}}{\partial v}$
CFTER		Conversion constant (feet/earth radii)
TLIST		Trajectory integration list (See TRAJ subroutine)

NH1	Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence
NH2	Pointer to location in variable storage of the 2nd altitude layer bounding the current region of influence

2. Calling sequence

—

C. Output

1. COMMON

A	Variable length trajectory array to be written on trajectory tape
A(1-9)	$x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}$
A(10-15)	$\frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}, \frac{\partial \dot{x}}{\partial \alpha}, \frac{\partial \dot{y}}{\partial \alpha}, \frac{\partial \dot{z}}{\partial \alpha}$
⋮	⋮
A(40-45)	$\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}, \frac{\partial \dot{x}}{\partial v}, \frac{\partial \dot{y}}{\partial v}, \frac{\partial \dot{z}}{\partial v}$
A(46-51)	$\frac{\partial x}{\partial \lambda_i}, \frac{\partial y}{\partial \lambda_i}, \frac{\partial z}{\partial \lambda_i}, \frac{\partial \dot{x}}{\partial \lambda_i}, \frac{\partial \dot{y}}{\partial \lambda_i}, \frac{\partial \dot{z}}{\partial \lambda_i}$
⋮	⋮
A(6*NDPR+4 to 6*NDPR+9)	$\frac{\partial x}{\partial \lambda_{NLAMS}}, \frac{\partial y}{\partial \lambda_{NLAMS}}, \frac{\partial z}{\partial \lambda_{NLAMS}}, \frac{\partial \dot{x}}{\partial \lambda_{NLAMS}}, \frac{\partial \dot{y}}{\partial \lambda_{NLAMS}}, \frac{\partial \dot{z}}{\partial \lambda_{NLAMS}}$
	$\frac{\partial \ddot{x}}{\partial \alpha}, \frac{\partial \ddot{y}}{\partial \alpha}, \frac{\partial \ddot{z}}{\partial \alpha},$
	⋮
	$\frac{\partial \ddot{x}}{\partial v}, \frac{\partial \ddot{y}}{\partial v}, \frac{\partial \ddot{z}}{\partial v}$

Present only if λ's are present in the solution vector

$$\left.\begin{array}{ccc} \frac{\partial \ddot{x}}{\partial \lambda_i}, & \frac{\partial \ddot{y}}{\partial \lambda_i}, & \frac{\partial \ddot{z}}{\partial \lambda_i} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \frac{\partial \ddot{x}}{\partial \lambda_{NLAMS}}, & \frac{\partial \ddot{y}}{\partial \lambda_{NLAMS}}, & \frac{\partial \ddot{z}}{\partial \lambda_{NLAMS}} \end{array}\right\}$$ Present only if λ's are in the solution vector

A(9*NDPR+7 to 9*NDPR+9)

If no drag parameters C_DA/m are to be solved for, the partials of position, velocity, and acceleration wrt λ_i $\left(\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z})}{\partial(\lambda_i)}\right)$ are not placed into the A array.

CDAD2M — Current $\left(\frac{C_DA}{2m}\right)\frac{(ft^2)}{slug}$ in internal units

TALT — Current altitude of vehicle (ft.)

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

TRJTAP — Writes one record of binary trajectory information on the trajectory tape

SUBROUTINE IDENTIFICATION

A. Title

TRJPRO

B. Segment

NRTPOD

C. Called by Subroutine

NRTPOD

FUNCTION

Main driver controlling the coordination of all activities involving the trajectory segment, curve fit segment, and the trajectory print and update segment.

USAGE

A. Calling sequence

Call TRJPRO

B. Input

1. COMMON

KONTRL	Flag indicating mode of NRTPOD KONTRL = 1 Curve fit and trajectory = 2 Trajectory only
DCFLG	JDC card options (card column 41-50)
IFTEX	Exit flag from subroutine FIT IFTEX = 1 Solution has converged = 2 Maximum iterations exceeded and converging = 3 Failed K BOUNDS/8 = 4 Normal return = 5 Maximum iterations exceeded and converging
PSTFLG	JDC options (card columns 51-60)
PR2DPI	8 x 3 array of partial derivatives of accelerations with respect to ADBARV and the two drag layers of current influence

2. Calling sequence

—

C. Output

1. COMMON

—

2. Calling sequence

—

D. Error/action messages

SUBROUTINES USED

A. Library

—

B. Program

TPRLM	Performs necessary initialization prior to a differential correction pass
TRJGEN	Driver for the trajectory segment; generates the trajectory tape
DCITER	Driver for the curve fit segment
PRUDRV	Trajectory print and update driver

SUBROUTINE IDENTIFICATION

A. Title

TRJTAP

B. Segment

NRTPOD

C. Called by subroutine

TRJGEN

FUNCTION

Function is to write the trajectory tape used by the curve fit and trajectory print and update segments.

USAGE

A. Calling sequence

Call TRJTAP (IOPT)

B. Input

1. COMMON

ITRJTP	Trajectory tape number
TG	Integration time to go ... minutes from 0 hours day of epoch
TRAJX	Integration coordinates at time TG: position, velocity, acceleration, partials of position and velocity w. r. t. the category 1 variables
N	Total number of words in the A array to be written
TG	Trajectory time that the trajectory tape record is referenced to
TCRASH	Impact flag
	= 0 vehicle has not impacted
	$\neq$ 0 vehicle has impacted

CDAD2M	C_DA/m (ft^2/slug) drag parameter referenced to time TG
TRHOA	ρ (slug/ft^3) atmospheric density at time TG

2. Calling sequence

IOPT	Flag indicating type of trajectory record written on trajectory tape
	IOPT = 1 Writes a standard data record
	= 2 Writes a pseudo "end of file" record

C. Output

1. COMMON

—

2. Calling sequence

—

3. Trajectory tape

A(I)	Variable length trajectory array
	I = 1, . . . , N

where

A (1-9)	$x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}$
(10-15)	$\frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}, \frac{\partial \dot{x}}{\partial \alpha}, \frac{\partial \dot{y}}{\partial \alpha}, \frac{\partial \dot{z}}{\partial \alpha}$
	$\vdots \qquad \vdots \qquad \vdots$
(40-45)	$\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}, \frac{\partial \dot{x}}{\partial v}, \frac{\partial \dot{y}}{\partial v}, \frac{\partial \dot{z}}{\partial v}$

(46 to 6*NDPR+9)

$$\frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1}, \frac{\partial \dot{x}}{\partial \lambda_1}, \frac{\partial \dot{y}}{\partial \lambda_1}, \frac{\partial \dot{z}}{\partial \lambda_1}$$

$$\vdots$$

$$\frac{\partial x}{\partial \lambda_{NLAMS}}, \frac{\partial y}{\partial \lambda_{NLAMS}}, \frac{\partial z}{\partial \lambda_{NLAMS}},$$

$$\frac{\partial \dot{x}}{\partial \lambda_{NLAMS}}, \frac{\partial \dot{y}}{\partial \lambda_{NLAMS}}, \frac{\partial \dot{z}}{\partial \lambda_{NLAMS}}$$

(6*NDPR+10 to 9*NDPR+9)

$$\frac{\partial \ddot{x}}{\partial \alpha}, \frac{\partial \ddot{y}}{\partial \alpha}, \frac{\partial \ddot{z}}{\partial \alpha}$$

$$\vdots \quad \vdots \quad \vdots$$

$$\frac{\partial \ddot{x}}{\partial v}, \frac{\partial \ddot{y}}{\partial v}, \frac{\partial \ddot{z}}{\partial v}$$

$$\frac{\partial \ddot{x}}{\partial \lambda_1}, \frac{\partial \ddot{y}}{\partial \lambda_1}, \frac{\partial \ddot{z}}{\partial \lambda_1}$$

$$\vdots \quad \vdots \quad \vdots$$

$$\frac{\partial \ddot{x}}{\partial \lambda_{NLAMS}}, \frac{\partial \ddot{y}}{\partial \lambda_{NLAMS}}, \frac{\partial \ddot{z}}{\partial \lambda_{NLAMS}}$$

and NLAMS is the total number of $C_D A/m$'s appearing in the solution vector.

D. Error/action messages

SUBROUTINES USED

A. Library

—

B. Program

—

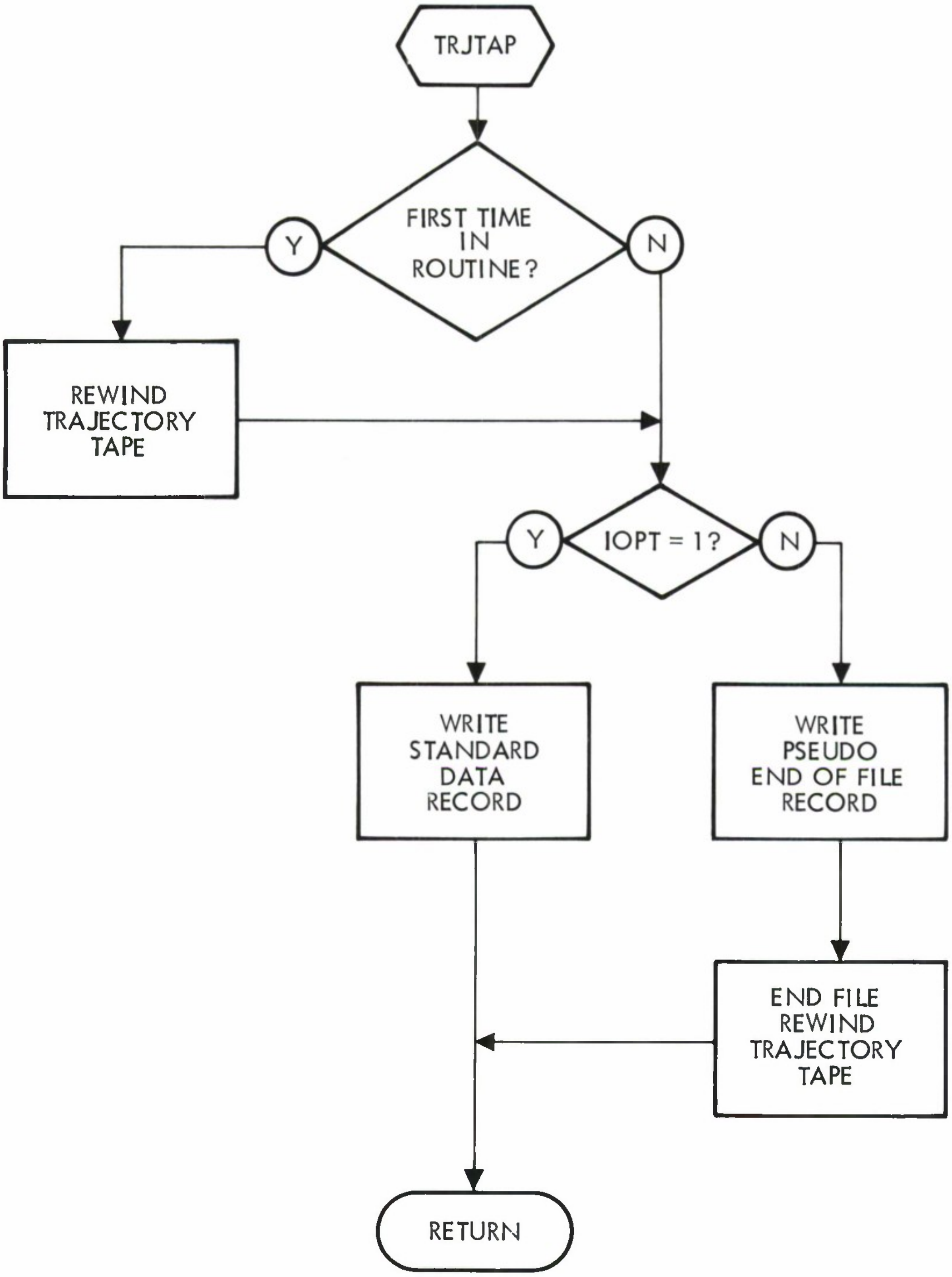


Figure 5-14. TRJTAP Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title
UPPER

B. Segment
NRTPOD

C. Called by subroutine
PRAUPD

FUNCTION

UPPER converts an N x N matrix stored lower triangular by rows into an N x N matrix stored upper triangular by rows with an augmented column. (This matrix with the augmented column is required as an input to subroutine LEGS2.)

USAGE

A. Calling sequence

Call UPPER (B, N)

B. Input

1. COMMON

NATA	Starting location in variable storage of the accumulated normal matrix.

2. Calling sequence

B	Input lower triangular matrix stored by rows
N	Dimension of matrix [B] is N x N

C. Output

1. COMMON

VSTR(NATA)	Output upper triangular matrix with an augmented column stored by rows

2. Calling sequence

—

D. Error/action messages

—

UPPER

SUBROUTINES USED

A. Library

—

B. Program

—

6. NRTPD2 STORAGE MAP

6.1 NEW/REVISED COMMON ARRAYS

The following labeled COMMON blocks have been introduced or revised.

1) COMMON/SIGBUF/SBUF(400)

   This COMMON array is used in the differential correction link as a buffer for the computed functional standard deviations. These standard deviations are computed in parallel with the observation residuals, are buffered in the SBUF array, and printed a page at a time.

2) COMMON/PLSS/PWPP(63), PWDTPP (63), PWDT2P (63)

   The PWPP, PWDTPP, PWDT2P arrays have been taken out of PLS labeled COMMON and placed into PLSS labeled COMMON. These arrays have been extended from 24 cells to 63 cells respectively. (See Reference 2, Page 2-45.)

$$\text{PWPP}(1 - 3) = \frac{\partial w_1}{\partial p_1}, \frac{\partial w_2}{\partial p_1}, \frac{\partial w_3}{\partial p_1}$$

$$(4 - 6) = \frac{\partial w_1}{\partial p_2}, \frac{\partial w_2}{\partial p_2}, \frac{\partial w_3}{\partial p_2}$$

$$\begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix}$$

$$\text{PWPP}\Big[3 + 3(n - 1)\Big] = \frac{\partial w_1}{\partial p_n}, \frac{\partial w_2}{\partial p_n}, \frac{\partial w_3}{\partial p_n}$$

where

$$\text{PWPP}\Big[3 + 3(n - 1)\Big] = \frac{\partial w_3}{\partial p_n}$$

and n is the number of parameters p to be solved for from the list (α_o, δ_o, B_o, A_o, R_o, V_o, λ_1, λ_2, $\cdots$, λ_{15}), and $\overline{W} = (w_1, w_2, w_3)$ is the geocentric position vector of the vehicle in a station meridian equatorial system.

$$PWDTPP(1,2,3) = \frac{\partial \dot{w}_1}{\partial p_1}, \frac{\partial \dot{w}_2}{\partial p_1}, \frac{\partial \dot{w}_3}{\partial p_1}$$

$$\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array}$$

$$PWDTPP\left[3 + 3(n - 1)\right] = \frac{\partial \dot{w}_1}{\partial p_n}, \frac{\partial \dot{w}_2}{\partial p_n}, \frac{\partial \dot{w}_3}{\partial p_n}$$

where n is defined as above and $\dot{\overline{W}} = (\dot{w}_1, \dot{w}_2, \dot{w}_3)$ is the geocentric earth fixed velocity vector of the vehicle in a station meridian equatorial system.

$$PWDT2P(1,2,3) = \frac{\partial \ddot{w}_1}{\partial p_1}, \frac{\partial \ddot{w}_2}{\partial p_1}, \frac{\partial \ddot{w}_3}{\partial p_1}$$

$$\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array}$$

$$PWDT2P\left[3 + 3(n - 1)\right] = \frac{\partial \ddot{w}_1}{\partial p_n}, \frac{\partial \ddot{w}_2}{\partial p_n}, \frac{\partial \ddot{w}_3}{\partial p_n}$$

where n is defined as above and $\ddot{\overline{W}} = (\ddot{w}_1, \ddot{w}_2, \ddot{w}_3)$ is the geocentric earth fixed acceleration of the vehicle in a station meridian equatorial system.

3) <u>COMMON/PLS/PLS(150)</u>

Although the PLS labeled COMMON region has been increased in size from 125 to 150, no new cells were introduced leaving cells PLS(117) to PLS(150) inclusive, unused in the differential correction link.

4) <u>COMMON/SMATRX/SMAT(630)</u>

The SMAT array has been removed from DATA COMMON storage (see Page 2-48, Reference 2), and placed into a labeled COMMON block, SMATRX. This array now allows storage for a 35 x 35 upper triangular by rows a priori normal matrix (A^TA). This storage was extended from a maximum allowable 20 x 20 to accommodate up to 15 drag layers in the solution vector.

5) COMMON/UMATRX/UPMAT(231)

The UPMAT array has been removed from DATA storage (see Page 2-50, Reference 2), and placed into labeled COMMON block, UMATRX. This array now allows storage for up to a 21 x 21 lower triangular covariance matrix stored by rows. This storage was extended from a maximum allowable 7 x 7 to accommodate up to 15 drag layers plus 6 initial conditions in the a priori covariance matrix.

6) COMMON/INPP/DTMP(690), DATA(1250)

Array DTMP has increased from 300 to 690 to accommodate up to 20 sensors of information. DTMP cells (1-50) contain the same information as before (see Page 2-46, Reference 2). DTMP cells (51-690) contain the following information per each sensor:

DTMP	(51)	Station ID	
	(52)	Range bias (km)	
	(53)	Azimuth bias (deg)	
	(54)	Elevation bias (deg)	
	(55)	Range rate bias (km/sec)	
	(56)	Range acceleration bias (m/sec^2)	
	(57)	Time bias (sec)	
	(58)	σ_R standard deviation on range (km)	
	(59)	σ_A standard deviation on azimuth (deg)	
	(60)	σ_E standard deviation on elevation (deg)	
	(61)	$\sigma_{\dot{R}}$ standard deviation range rate (km/sec)	
	(62)	$\sigma_{\ddot{R}}$ standard deviation range acceleration (m/sec^2)	
	(63)	B_R	
	(64)	B_A	
	(65)	B_E	
	(66)	$B_{\dot{R}}$	
	(67)	$B_{\ddot{R}}$	
	(68)	θ_1	
	(69)	θ_2	
	(70)	θ_3	Sensor data input associated
	(71)	θ_4	with the functional standard
	(72)	θ_5	deviation option. See
	(73)	θ_6	Section 2.2.1
	(74)	θ_7	
	(75)	$f(\theta_1)$	
	(76)	$f(\theta_2)$	
	(77)	$f(\theta_3)$	
	(78)	$f(\theta_4)$	
	(79)	$f(\theta_5)$	
	(80)	$f(\theta_6)$	
	(81)	$f(\theta_7)$	
	(82)	blank	

The above information associated with each sensor follows in the DTMP array. Since 20 sensors are allowed and 32 cells are reserved in the DTMP array for each sensor, 690 cells have been allotted to DTMP.

The new cells added to DATA storage are in Section 6.2.

7) COMMON/PRDD/PR2DPI(24)

The PR2DPI array has been introduced to contain the

$$\frac{\partial \vec{\ddot{R}}}{\partial(\text{CAT } 1)}$$

partial derivatives. These partial derivatives are computed at each integration step and are stored in the following manner

$$\text{PR2DPI}(1-3) = \frac{\partial \ddot{x}}{\partial \alpha}, \quad \frac{\partial \ddot{y}}{\partial \alpha}, \quad \frac{\partial \ddot{z}}{\partial \alpha}$$

$$(4-6) = \frac{\partial \ddot{x}}{\partial \delta} \quad \frac{\partial \ddot{y}}{\partial \delta}, \quad \frac{\partial \ddot{z}}{\partial \delta}$$

$$\begin{matrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{matrix}$$

$$(16-18) \quad \frac{\partial \ddot{x}}{\partial v}, \quad \frac{\partial \ddot{y}}{\partial v}, \quad \frac{\partial \ddot{z}}{\partial v}$$

where λ_i, λ_{i+1} are the current $(C_D A/m)$ drag parameters bounding the region of influence

$$(19-21) = \frac{\partial \ddot{x}}{\partial \lambda_i}, \quad \frac{\partial \ddot{y}}{\partial \lambda_i}, \quad \frac{\partial \ddot{z}}{\partial \lambda_i},$$

$$(22-24) = \frac{\partial \ddot{x}}{\partial \lambda_{i+1}}, \quad \frac{\partial \ddot{y}}{\partial \lambda_{i+1}}, \quad \frac{\partial \ddot{z}}{\partial \lambda_{i+1}},$$

8) COMMON/TRJX/TRAJX(198) or A(198)

The arrays TRAJX and A above are identical arrays with different names. The A array is used in the subroutines associated with the writing of the trajectory tape and the TRAJX array is used elsewhere. (See subroutine TRJOUT, Section 5.4 for a detailed description of this array).

This labeled COMMON block was added to accommodate the extended partial derivative arrays (see subroutine TRJOUT) associated with the additional 15 drag layers which may appear in the solution vector.

9) COMMON/TDPD/TDPDX(441), FINK(441), DINK(441), RINK(441)

This labeled COMMON block was introduced in the "UPDATE" link to be used as temporary working storage. The arrays defined by TDPD contain at different times during the "Updating," the partials of polar coordinates with respect to cartesian coordinates; the sigma and rho matrix (polar); the normal matrix (polar); the cartesian covariance matrix; etc.

6.2 NEW CELLS ADDED TO DATA STORAGE

Name	Equivalence	Dimension	Description	
BIJ	954	100	b_{ij} constraint matrix	See Section 4, Operational Considerations, in Reference 8.
XIJ	1054	100	i* 100 + j for b_{ij}	
CI	1154	30	additive constants for linear constraints	
NMDTMP	1184	1	Number of cells per sensor in DTMP list.	
ALTS	1185	15	Altitude table for multiple drag (kilometers)	
CLAMDA	1200	15	$C_D A/M$ table corresponding to ALTS table (meters^2/kilogram).	
CATLM	1215	15	The CATLM array indicates to the program the CLAMDA variables to be solved for. This array must contain either "ones" or "zeros," a 1 indicating the corresponding variable is to be solved for. For example, to solve for the second, fourth, and fifth CLAMDA, the ALTS-CLAMDA arrays must contain at least 5 entries, and the CATLM array must be input: CATLM = 0, 1, 0, 1, 1,	

6.3 BLANK COMMON

1) Array BLK 1 additions:

Name	BLK1( )	Dimension	Description
CHEPS	BLK1(40)	1	Epsilon on altitude cut-offs. An altitude layer is accepted as a cut-off point when the computed cut-off altitude, H_C, is within CHEPS of the input cut-off altitude, H_I. Or the computed cut-off altitude, H_c, is accepted when the following condition exists: $\left\| H_c - H_I \right\| \leq$ CHEPS. CHEPS is nominally set at 10^{-6} earth radii.

2) Array BLK2 has been increased from 30 to 50 cells. The following variables have been added:

Name	BLK2( )	Dimension	Description
MPR	23	1	Size of the constrained system (when using linear constraints).
MBNDS	24	1	Starting location in VSTR of the bounds vector for the constrained system.
IMAX	25	1	The number of non-zero elements of b_{ij} (linear constraints).
NIJ	26	1	Starting location in VSTR of the vector containing the i* 100 + j where i, j refer to the elements of b_{ij}.
NST	27	1	Temporary storage used in VSTR for constraining the size of the system from N to M.
NB	28	1	VSTR pointer for non-zero elements of the constraint matrix.
NC	29	1	Starting location in VSTR for additive constants used in linear constraints.
NMSTAT	30	1	Number of cells allotted per station in the master sensor table.

Name	BLK2( )	Dimension	Description
NLAM	31	1	Total number of entries in the altitude C_DA/m table.
NLAMS	32	1	Number of drag parameters C_DA/m in the solution vector.
NH	33		Starting location in VSTR of the altitude C_DA/m table.
NLID	34		Starting location in VSTR of the identifiers for the drag parameters C_DA/m appearing in the solution vector.
NPALM	35	1	Starting location in VSTR of the b_i vectors where

$$b_i = \left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_t}{\partial(\alpha, \delta, \beta, A, R, v)_{t_o}}\right]^{-1}$$

$$\times \left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_t}{\partial\left(\frac{C_DA}{m}\right)_i}\right]$$

$$= \left[\frac{\partial(\alpha, \delta, \beta, A, R, v)_{t_o}}{\partial\left(\frac{C_DA}{m}\right)_i}\right]$$

$$i = 1, \cdot \cdot \cdot, \text{NLAM}$$

This array is NLAM x 6 cells long and is stored in the following manner:

$$\text{VSTR(NPALM to NPALM+5)} = \frac{\partial\alpha}{\partial\lambda_1}, \frac{\partial\delta}{\partial\lambda_1}, \frac{\partial\beta}{\partial\lambda_1}, \frac{\partial A}{\partial\lambda_1}, \frac{\partial R}{\partial\lambda_1}, \frac{\partial v}{\partial\lambda_1}$$

$$\cdot \qquad \frac{\partial\alpha}{\partial\lambda_2}, \cdot \quad \cdot \quad \cdot \quad \cdot \quad \frac{\partial v}{\partial\lambda_2}$$

$$\cdot \qquad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\cdot \qquad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\text{VSTR(NPALM+6*NLAM-1)} = \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \frac{\partial v}{\partial\lambda_{NLAM}}$$

Name	BLK2( )	Dimension	Description

where

$$\lambda_i = \left(\frac{C_D A}{m}\right)_i$$

NPXLM 36 1 Starting location in VSTR of the c_i vectors, where

$$c_i = \left[\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_t}{\partial\left(\frac{C_D A}{m}\right)_i}\right]$$

$$i=1, \cdot \cdot \cdot, \text{NLAM}$$

This array is NLAM x 6 cells long and is stored in the following manner:

$$\text{VSTR(NPXLM to NPXLM+5)} = \frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1}, \frac{\partial \dot{x}}{\partial \lambda_1}, \frac{\partial \dot{y}}{\partial \lambda_1}, \frac{\partial \dot{z}}{\partial \lambda_1}$$

$$\text{(NPXLM+6 to NPXLM+11)} = \frac{\partial x}{\partial \lambda_2}, \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \frac{\partial \dot{z}}{\partial \lambda_2}$$

$$\cdot \qquad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\cdot \qquad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\text{VSTR(NPXLM+6*NLAM-1)} = \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \frac{\partial \dot{z}}{\partial \lambda_{\text{NLAM}}}$$

where

$$\lambda_i = \left(\frac{C_D A}{m}\right)_i$$

NPXDLM 37 1 Starting location in VSTR of the d_i vectors, where

$$d_i = \left[\frac{\partial(\ddot{x}, \ddot{y}, \ddot{z})_t}{\partial\left(\frac{C_D A}{m}\right)_i}\right]$$

$$i=1, \cdot \cdot \cdot, \text{NLAM}$$

Name	BLK2( )	Dimension	Description
			This array is NLAM x 3 cells in length and is stored in the following manner:

$$\text{VSTR(NPXDLM to NPXDLM+2)} = \frac{\partial \ddot{x}}{\partial \lambda_1}, \frac{\partial \ddot{y}}{\partial \lambda_1}, \frac{\partial \ddot{z}}{\partial \lambda_1}$$

$$\text{VSTR(NPXDLM+3 to NPXDLM+5)} = \frac{\partial \ddot{x}}{\partial \lambda_2}, \frac{\partial \ddot{y}}{\partial \lambda_2}, \frac{\partial \ddot{z}}{\partial \lambda_2}$$

$$\vdots \qquad\qquad \vdots \quad \vdots \quad \vdots$$

$$\text{VSTR(NPXDLM+3*NLAM-1)} = \cdot \quad \cdot \quad \frac{\partial \ddot{z}}{\partial \lambda_{NLAM}}$$

where

$$\lambda_i = \left(\frac{C_D A}{m}\right)_i$$

3) Array BLK3

Name	BLK3( )	Dimension	Description
ECRIT	90	1	Minimum elevation at which steering ephemeris is allowed. Nominally set at -5°. (See Page 3-5, Steering Ephemeris.)
RDFLG	91	1	Flag which controls the output units of range rate and range acceleration in the steering ephemeris. Nominally set = 0. (See Page 3-5, Steering Ephemeris for further description.)

4) Array BLK 4 has been increased from 400 to 450 cells. The following variables have been added:

Name	BLK4( )	Dimension	Description
PRMS	392	5	The RMS from the previous iteration of the range, azimuth, elevation, range rate, and range acceleration residuals.

Name	BLK4( )	Dimension	Description
CFLAG	397	1	(See page 2-60 of Reference 1.)
IFIT	398	1	
CFLG	399	1	Linear constraints flag for additive constants.
IFVE	400	2	Two flags indicating whether the respective C_DA/m in the region of influence is in the solution vector or not. IFVE = 0 the C_DA/m of a particular region is not in the solution vector. IFVE ≠ 0 the C_DA/m of a particular region is in the solution vector.
ALT	402	2	Two altitude layers bounding the current region of influence. (e. r.)
INFG	404	1	Flag indicating to subroutine PLAMDA whether an altitude crossing has occurred and which region of drag influence has been entered. INFG = 0 No altitude crossing has occurred INFG = 1 Vehicle has re-entered altitude region 1, i. e., has crossed ALT(1). INFG = 2 Vehicle has left the influence of region 1 and crossed into altitude region 2.
NDPRT	405	1	Number of CAT1 variables plus number of C_DA/m drag parameters being integrated at any one instance (either 6 or 8).
NH1	406	1	Location in VSTR of the 1st altitude layer bounding the current region of influence.
NH2	407	1	Location in VSTR of the 2nd altitude layer bounding the current region of influence.

7. PREPCD

PREPOD is a preliminary orbit determination program designed to be used in conjunction with the NRTPOD program. A preliminary estimate of the position and velocity of a satellite is derived by fitting an orthogonal polynomial of degree ≤ 4 in the least squares sense to the components of n position fixes (topocentric range, elevation, azimuth) from an observing sensor. The velocity components are obtained by differentiating the position polynomials (R, A, E) with respect to time.

The observations, station biases, if any, and station coordinates are input on cards in the NRTPOD format. See Reference 1 for a description of these input cards. The degree of the polynomial fit is nominally set to four and is automatically adjusted downward if there are fewer than five observations. Also, the degree may be selected by the analyst, provided that the number of observations exceeds the degree by at least one.

The epoch may be selected at any time past (and including) the time of the first observation. If epoch is not specified or if epoch is specified prior to the time of the first observation, the time of the first observation is automatically selected as epoch.

The evaluated polynomials of the topocentric quantities (R, A, E, $\dot{R}$, $\dot{A}$, $\dot{E}$) are converted to Cartesian Earth-centered inertial coordinates. For output purposes, the geocentric Cartesian coordinates are transformed to polar spherical (ADBARV) coordinates, if desired.

The output consists of the geocentric inertial state vector and the associated epoch time; it is printed and punched on cards which are suitable for input to NRTPOD.

The flow diagram on the following page is a general computer program logic flow of PREPOD. The following sections describe the input/output, COMMON storage, and subroutines. A description of the mathematical techniques used in PREPOD to fit an orthogonal polynomial to n observations in a least squares sense is given in Reference 9.

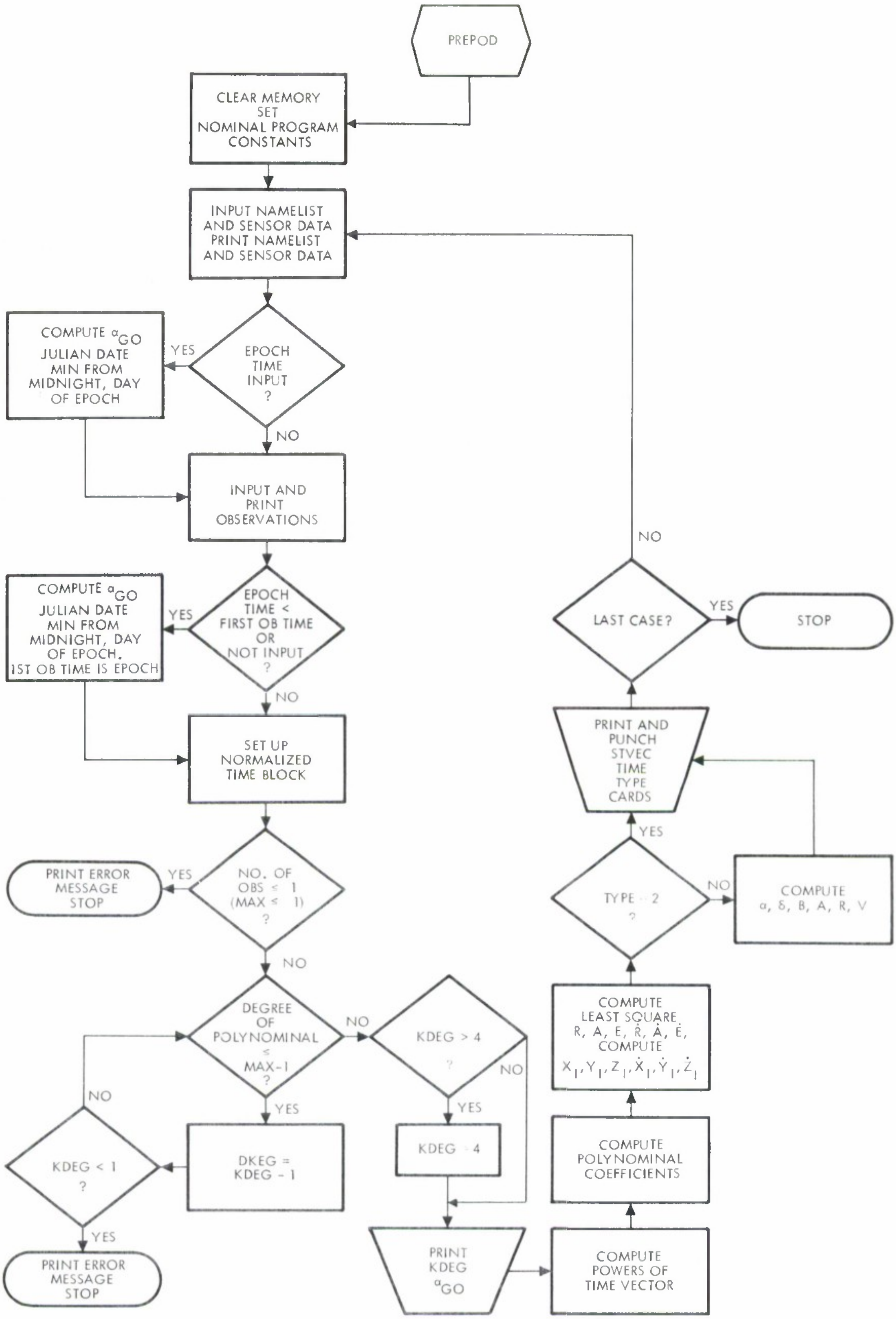


Figure 7-1. PREPOD Flow Diagram

The formulation of the least squares curve fit as programmed in PREPOD could be easily modified to make it a weighted least squares curve fit. In Figure 7-5 the provision for printing the standard deviation for each observation is shown, should this modification be implemented.

7.1 PROGRAM INPUT

7.1.1 Deck Set-Up

The input deck consists of NAMELIST cards (program options), sensor cards, and observation cards, in that order. Cases may be stacked although some care must be exercised when program constants are changed from their nominal values. Program constants which are changed by NAMELIST are not restored to the nominal value after each case; hence, nominal values must be restored by the user in the subsequent cases of a stacked run. The deck set-up is shown pictorially in Figure 7-2.

When the program is run without any options, the following nominal program conditions prevail:

1) The polynomial fit is of degree 4.
2) The preliminary estimate is output in polar spherical coordinates.
3) The epoch is selected at the time of the first observation.
4) The observations must be presorted.

7.1.2 Input Cards

There are three basic types of input cards for the PREPOD program:

NAMELIST inputs
Sensor cards
Observation cards

NAMELIST — The NAMELIST inputs constitute the preliminary data cards that specify the program options and the constants. The various types of cards are explained below. Some remarks concerning the use of the NAMELIST format are given in Reference 1, Section 1.2.4.

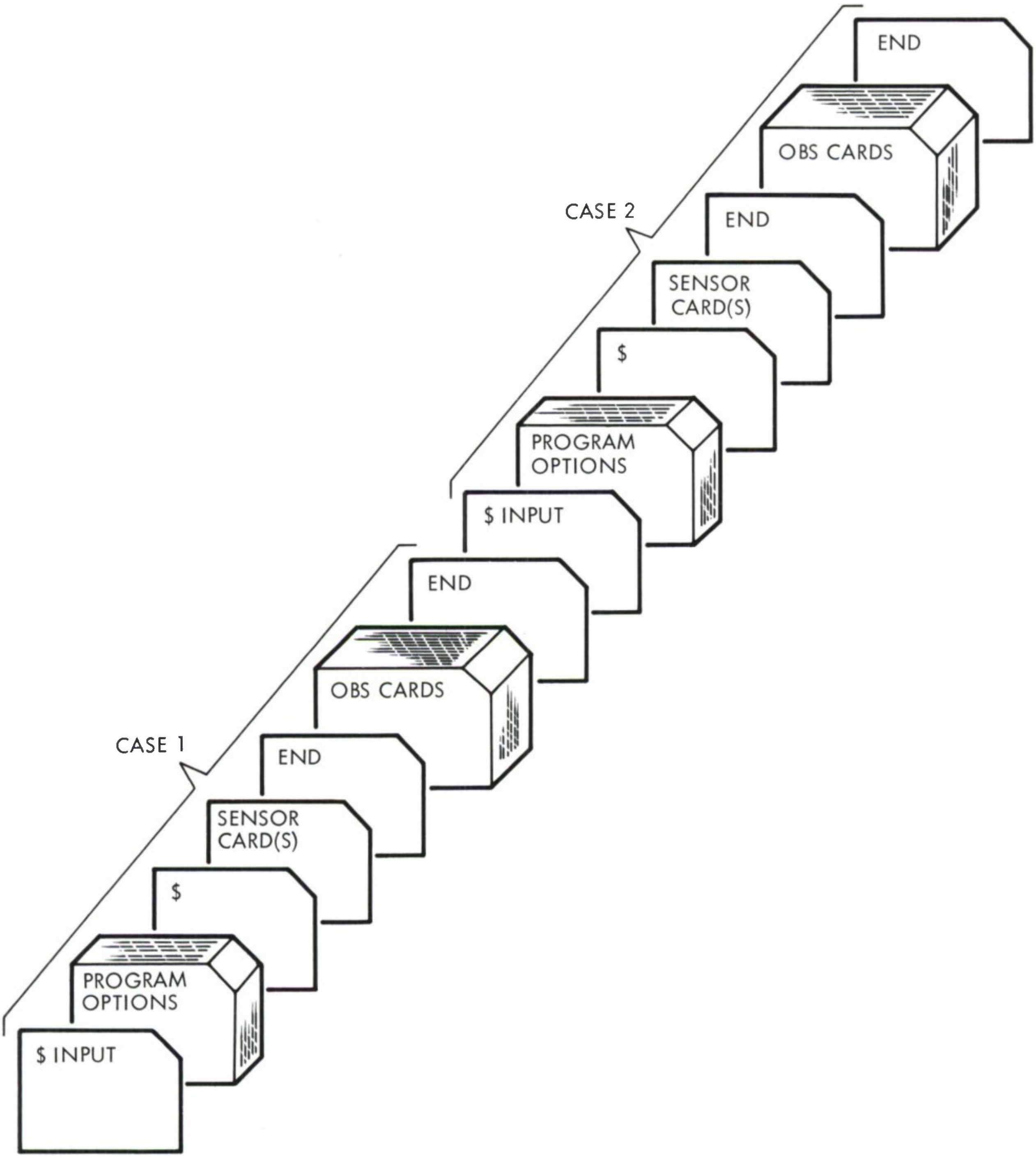


Figure 7-2. PREPOD Input Deck

KDEG — The KDEG card specifies the degree of the polynomial fit. If this card is omitted from the input deck, the degree is set to 4. And if there are fewer than five observation cards, KDEG is automatically adjusted to be equal to one less than the number of observations. The maximum permissible value for KDEG is 4. This input is a single entry array.

Example

KDEG = 3.,

TIME — This input permits the analyst to select the epoch to which the derived preliminary estimate is to be referenced. If this card is omitted from the input deck, the epoch is selected as the time of the earliest observation. If the selected time precedes the first observation, the program automatically resets the epoch to the time of the first observation. The TIME card is a six-entry array as follows:

TIME = Year-1900, Month number, day number,
hours, minutes, seconds,

Example

TIME = 66, 10, 17, 14, 48, 21. 78,

TYPE — This card specifies the coordinate system of the preliminary estimate of the orbit. If TYPE = 1, the output state vector is in polar spherical coordinates (ADBARV). And if TYPE = 2, the output is in Cartesian components of position and velocity. In both cases, the reference direction is the vernal equinox and the reference plane is the equator; and both coordinate systems are earth-centered inertial (ECI).

PRFLG5 — This input is a flag which indicates if the observations are presorted. The flag is nominally zero, which indicates that the observation cards are in time sort, If PRFLG5 = 1, the observations processor will sort the cards. (Maximum = 300 cards).

CONS — The CONS input is used to change the constants of the program. The program constants are printed in a block at the beginning of each run. The identification of each constant with respect to program location is given in Section 7.3. The constants which are changed with a CONS card are not restored to the nominal value after each case; therefore, nominal values must be restored by the analyst if stacked cases are submitted on one computer run.

Example

CONS(3) = 0.437528E-2,

Sensor Cards	PREPOD uses the standard NRTPOD sensor cards as described in Reference 1, Section 1.2.5. However, the program only accepts sensor cards Type 1 and 2, station coordinates and observable biases respectively. Since the PREPOD curve fit is not weighted, standard deviations (type = 3) are not required. If a sensor bias card is input, the biases are removed from the observations before normal observation processing begins. It should be noted that there are two sensor cards at most in any one determination since PREPOD derives a preliminary estimate from the observations of one sensor only.
Observation Cards	PREPOD uses the standard NRTPOD observation cards as described in Reference 1, Section 1.2.6. The program accepts range, azimuth, and elevation observables only; however, if range-rate or range-acceleration observables are input also, the program ignores them. The maximum number of observations the program will handle is 300. The observations do not have to be time sorted; however, an additional NAMELIST card (PRFLG5) must be input if the cards are presorted.

Figure 7-3 is a sample input sheet for two cases. The first case consists of nominal inputs only; in the second case, some of the program options are called. The parenthetical information of Figure 7-3 does not constitute part of the input; it is explanatory only.

7.2 PROGRAM OUTPUT

PREPOD produces printed output and punched output on cards suitable for input to NRTPOD.

7.2.1 Printed Output

The printed output is sectioned into one or more pages of particular information. The sections are outlined below in the output sequence.

Data	No. of Pages
NAMELIST input	1
Constants, sensors, and observations	1 or more
Initial Conditions	1

NAMELIST Input	The NAMELIST input print is essentially a card image listing of the input NAMELIST cards. Figure 7-4 is a sample input listing of PREPOD.
Constants, Sensors, and Observations	Following the NAMELIST input listing is a listing of the remaining input cards and the program constants. Figure 7-5 is a sample of this output. The program constants /BLK1/CONS(30) are printed at the top of the page. The definition of these constants is given in Section 7.3. Following the program constants is a listing of the sensor ID, location, and biases, if any. The observations are listed following the sensor information. Note that provision has been made for the standard deviation of each observable. Following the observations print is a message indicating the degree of the polynomial fit and the value of α_{go}, the right ascension of Greenwich at 0.0 hrs. on the day of epoch.
Initial Conditions	The last page of output consists of the derived preliminary estimate of the orbit and its associated epoch. The type of coordinate system in which the estimate is output is also indicated. Figure 7-6 is a sample printout of the Initial Conditions page. The output units are degrees, kilometers, and kilometers/second.

7.2.2 Punched Output

The punched output of PREPOD is in the proper format for input to NRTPOD. The three output variables are listed below:

Output Variable	Definition
STVEC	The derived initial conditions in either polar spherical or Certesian coordinates
TIME	The epoch date and time associated with the initial conditions
TYPE	The type of coordinate system; TYPE = 1, Polar Spherical Coordinates TYPE = 2, Cartesian Coordinates

NAME ____ PROBLEM NO. ____ NO. OF CARDS ____

COL N FREE KEY PUNCH FORM

PRIORITY ____ KEYPUNCHED BY ____ VERIFIED BY ____

```
$INPUT                    (NOMINAL CASE)
$
MH    1  42.61730  288.50860     156.                                (SENSOR)
END
MH    6510  1  133  0.999985026 5.4197      10.6817        9996.605       -1.3237   (OBS)
MH    6510  1  134  0.999985026 7.1456      11.5001        9918.420       -1.2823
MH    6510  1  135  0.999985026 8.9048      12.3007        9842.755       -1.2396
MH    6510  1  136  0.999985027 0.6976      13.0822        9769.695       -1.1955
END
$INPUT                 (EPOCH TIME, TYPE, AND KDEG INPUT)
      TYPE=2
      KDEG=2
      TIME=65,10,1,1,34,0.999985
$
MH    1  42.61730  288.50860     156.                                (SENSOR)
MH    2     1.0        1.0          1.0                              (SENSOR BIAS)
END
MH    6510  1  133  0.999985026 5.4197      10.6817        9996.605       -1.3237   (OBS)
MH    6510  1  134  0.999985026 7.1456      11.5001        9918.420       -1.2823
MH    6510  1  135  0.999985026 8.9048      12.3007        9842.755       -1.2396
MH    6510  1  136  0.999985027 0.6976      13.0822        9769.695       -1.1955
END
```

STL FORM 708 REV. 5-65

Figure 7-3. PREPOD Sample Input Sheet for Two Stacked Cases

```
$INPUT          FCC TEST CASE
     TYPE=2
     KDEG=2
     TIME=65,10,1,1,34,0.999985
```

Figure 7-4. Sample NAMELIST Input Listing

PRELIMINARY ORBIT DETERMINATION PROGRAM

PROGRAM CONSTANTS

1	0.63781650E 04	2	0.57295779E 02	3	0.43752650E-02	4	0.34566194E-02	5	0.
6	0.30999999E 02	7	0.27999999E 02	8	0.30999999E 02	9	0.30000000E 02	10	0.30999999E 02
11	0.30000000E 02	12	0.30999999E 02	13	0.30999999E 02	14	0.30000000E 02	15	0.30999999E 02
16	0.30000000E 02	17	0.30999999E 02	18	0.62831853E 01	19	0.31415925E 01	20	0.
21	0.	22	0.	23	0.	24	0.	25	0.30000000E 01
26	0.12000000E 02	27	0.20000000E 01	28	0.20000000E 01	29	0.20000000E 01	30	0.

SENSOR LOCATION

ID	LATITUDE	LONGITUDE	ALT	R BIAS	A BIAS	E BIAS	TIME BIAS
MH	42.6173	288.5086	156.	0.	0.	0.	0.

OBSERVATION TYPE

ID	T-T0	YR	MN	DY	HR	MIN	SECS	RANGE SIGMA R	AZ SIGMA A	EL SIGMA E	R DOT SIGMA R.
MH	-1.000	65	10	1	1	33	1.000	9997.	265.4197	10.6817	0.
								0.	0.	0.	0.
MH	0.	65	10	1	1	34	1.000	9918.	267.1456	11.5001	0.
								0.	0.	0.	0.
MH	1.000	65	10	1	1	35	1.000	9843.	268.9048	12.3007	0.
								0.	0.	0.	0.
MH	2.000	65	10	1	1	36	1.000	9770.	270.6976	13.0822	0.
								0.	0.	0.	0.

POLYNOMIAL FOR FITTING OBS IS OF DEGREE 2

ALPHA G ZERO = 9.5148 DEGREES

Figure 7-5. Sample Output of Program Constants, Sensor Location and Biases, and Observations

```
                         INITIAL CONDITIONS

                 YEAR MONTH DAY HOUR MINUTE SECOND

                  65    10    1    1    34    1.000

      X                Y                Z                XDOT              YDOT              ZDOT

-9.4969442E 02   -1.1634953E 04   5.2765009E 03   3.7164931E-01   2.2511337E 00   5.0287881E 00

                            TYPE = 2
```

Figure 7-6. Sample Initial Conditions Page

```
STVEC=-0.94969442E 03,-0.11634953E 05, 0.52785009E 04
       0.37164931E-00, 0.22511337E 01, 0.50287880E 01
TIME=65,10, 1, 1,34, 1.000
TYPE=2
```

Figure 7-7. Listing of Punched Card Output

Figure 7-7 is a card listing of the punched card output of PREPOD. These punched cards correspond to the sample printout case described in the previous section.

7.3 PREPOD STORAGE MAP

/BLK/CONS(30)	Program Constants	
CONS(1)	CKMER	Kilometers/earth radii
CONS(2)	CDEG	Degrees/radian
CONS(3)	CWE	ω_e, rotation rate
CONS(4)	CELLIP	f, ellipticity of earth
*CONS(5)	CAE	Internal units/earth radii
CONS(6)	CDAYMN(12)	Day of the month, non-leap year Jan = CDAYMN(1), etc.
CONS(18)	C2PI	2π
CONS(19)	CPI	π
CONS(25)	KOUT	System output tape
CONS(26)	IOUT	System punch tape
CONS(27)	KIN	System input tape
CONS(28)	ITYPE	Type of output 1 = ADBARV 2 = Cartesian
CONS(29)	KDEG	Degree of polynomial 4 = Nominal

COMMON/BLK2/WSTR(100)		
WSTR(1)	TIME(6)	Epoch time, year, month, day, hour, minutes, seconds
WSTR(7)	TEPOCH	Minutes from midnight, day of epoch
WSTR(8)	STVEC(6)	State vector ITYPE = 1 α, β, δ, A, R, V ITYPE = 2 $X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}$

*Not used when internal units are earth radii

WSTR(14)	ITFLG	Epoch time flag 1 = Epoch input 0 = First ob is epoch
WSTR(15)	PRFLG5	Sort obs flag 1 = Presorted 0 = sort
WSTR(16)	TJDATE	Julian date of midnite day of epoch
WSTR(17)	TALFAG	α_{g_o}
WSTR(18)	IFRST	First time into BCDOBS flag
WSTR(25)	STID	Station ID 3 Hollerith characters
WSTR(26)	STLAT	Station latitude
WSTR(27)	STLONG	Station longitude
WSTR(28)	STALT	Station altitude
WSTR(29)	ABIAS	Station azimuth bias
WSTR(30)	EBIAS	Station elevation bias
WSTR(31)	RBIAS	Station range bias
WSTR(32)	TBIAS	Station time bias
WSTR(33)	IBIAS	Bias flag
WSTR(41)	A(20)	Coefficients of least square fit polynomial
WSTR(81)	TP(20)	Powers of time vector

COMMON/TMHLD/TDJ(300)	Observation times
COMMON/OBHLD/OBSV(900)	Observations
OBSV(1)	R(300)
OBSV(301)	A(300)
OBSV(601)	E(300)
COMMON/BLK4/TMNER(300)	Normalized obs times
COMMON/TEMP/TEMP(59)	Temporary working storage

7.4 PREPOD SUBROUTINE DESCRIPTIONS

This section contains a glossary and description of the PREPOD program subroutines.

7.4.1 Subroutine Glossary

This glossary is an alphabetical list of the PREPOD subroutines. The subroutines having an asterisk denote the following documentation:

* NRTPOD documentation, Reference 1

** ESPOD documentation, Reference 2

Subroutine	Functional Description
BCDOBS	Reads one observation card.
CLTIME*	Computes Gregorian time.
CTOP**	Converts a Cartesian state vector to polar spherical coordinates (ADBARV).
DOT*	Computes scalar product.
INPTC	Reads NAMELIST input and NRTPOD sensor cards.
LODOBS	Main control for observation card processor.
LSQFIT	Computes the coefficients of an orthogonal polynomial for fitting observations in the least square sense.
MAGN**	Computes the magnitude and magnitude squared of a 3-D vector.
OBSIN	Scales observations to internal units and applies biases, if any.
OBSSRT	Sorts observations by time.
PIMOD*	Modulates an angle between 0 and 2π.
PNCHVC	Punches the PREPOD output in NAMELIST format.
PRNTC	Prints the block of program constants.
RADSQ*	Computes the magnitude and magnitude squared of a 3-D vector.
SCLOUT	Scales the ECI state vector from internal units to external units.
SETCON	Sets constants for program.

Subroutine	Functional Description
TIME**	Converts Y, M, D, H, M, S to Julian date: days + fraction.
TINIT**	Sets up initial time, computes α_{g_o}.
TPTOIN	Converts topocentric state vector to Cartesian (ECI) coordinates.

7.4.2 Subroutine Descriptions

SUBROUTINE IDENTIFICATION

A. Title
   BCDOBS

B. Segment
   PREPOD

C. Called by subroutine
   LODOBS

FUNCTION

To read one BCD observation card and store for processing by LODOBS. It also calls TINIT to compute α_{g_o} and epoch time from the first observation card if no epoch time is requested or if epoch time precedes the first observation.

USAGE

A. Calling sequence
   Call BCDOBS (A, SEOF)

B. Input

   1. COMMON

/BLK1/CONS(30)	
KOUT	Output tape number
KIN	Input tape number
/BLK2/WSTR(100)	
ITFLG	Flag to indicate if epoch time and α_{g_o} desired
IFRST	Flag to test if first observation card

   2. Calling sequence

   —

C. Output

   1. COMMON

/BLK2/WSTR(100)	(If no epoch time requested)

TIME	Six-element vector - YR, M, D, H, MIN, SEC
TEPOCH	Minutes from midnite, day of epoch
TALFAG	α_{g_o}

2. Calling Sequence

A	Sixteen element vector of observables and standard deviations
SEOF	Flag to indicate end of observation cards

D. Error/action message

"PROGRAM IGNORES TYPES 1 AND 2 OBSERVATION CARDS."

If there is a punch in column 20, card is ignored.

SUBROUTINES USED

A. Library

Input/output

B. Program

TINIT
TIME

SUBROUTINE IDENTIFICATION

A. Title
INPTC

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION

To read the NAMELIST input and the NRTPOD sensor card. Also the working storage (WSTR) is initialized for stacked cases.

USAGE

A. Calling sequence
Call INPTC

B. Input

1. COMMON

/BLK1/CONS(30)

KOUT	System output tape number
KIN	System input tape number

2. Calling sequence

—

C. Output

1. COMMON

/BLK1/CONS(30)

KDEG	Degree of polynomial
ITYPE	Type of output 1 = ADBARV 2 = $XYZ\dot{X}\dot{Y}\dot{Z}$

/BLK2/WSTR(100)

TIME(6)	Calendar date of epoch if entered
ITFLG	Flag to indicate if an epoch time was entered
STID	Station identification
STLAT	Station latitude
STLONG	Station longitude
STALT	Station altitude
ABIAS	Station azimuth bias
EBIAS	Station elevation bias
RBIAS	Station range bias
TBIAS	Station time bias
PRFLG5	Flag to indicate that observations are sorted
IFRST	Flag to indicate that the epoch time must be checked against the first observation time

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

Input/output

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title
LODOBS

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION

To read and sort the observation cards. This subroutine is essentially unchanged from the NRTPOD version as described in Section 5. 3 of Reference 1. The following characteristics distinguish the PREPOD version:

1. U(3500), the temporary storage used as a buffer for observations, is smaller and is used as working storage by the LSQFIT subroutine.

2. 300 is the maximum number of observations which may be processed.

3. No tape is written.

SUBROUTINE IDENTIFICATION

A. Title

LSQFIT

B. Segment

PREPOD

C. Called by subroutine

PREPOD

FUNCTION

To calculate orthogonal polynomial coefficients for fitting a polynomial to a number of observations in the least-square sense.

USAGE

A. Calling sequence

Call LSQFIT (MAX, KDEG, PDJ, TDJ, XDJ, BUFS, BMD, A)

B. Input

1. COMMON

—

2. Calling sequence

MAX	Number of observations
KDEG	Degree of polynomial
PDJ	(KDEG+1)*MAX locations of temporary storage
TDJ	Independent variable (normalized) MAX number
XDJ	Dependent variable MAX number
BUFS	(KDEG+1)*4 locations of temporary storage
BMD	((KDEG+2)*(KDEG+3))/2 locations of temporary storage

C. Output

1. COMMON

—

2. Calling sequence

A. $A(1) = a_o$ — KDEG+1 locations for the

$A(2) = a_1$ — least square polynomial coefficients. These must be divided by the normalizing number for the independent variable in the calling program as shown below.

.
.
.
.

$A(d+1) = a_d$

where

$$X = a_o + (a_1/(t_n-t_o))\, t + (a_2/(t_n-t_o)^2)t^2 + \ldots$$
$$+ (a_d/(t_n-t_o)^d)t^d$$

D. Error/action messages

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

Given M observations t_1, x_1; t_2, x_2; : ·· t_M, x_M a polynomial of degree d is to be fitted to these observations. First, a(d+1)xM matrix of orthogonal polynomials is formed recursively as follows:

The times are normalized

$$t_J' = \frac{t_J - t_1}{t_M - t_1}$$

$$P_1(t'_j) = 1 \qquad (j = 1, M) \text{ for all } P_j\text{'s}$$

$$P_2(t'_j) = (t'_j - u_1)P_1(t'_j)$$

$$P_3(t'_j) = (t'_j - u_2)P_2(t'_j) - V_2P_1(t'_j)$$

$$\vdots$$

$$P_{d+1}(t'_j) = (t'_j - u_d)P_d(t'_j) - V_dP_{d-1}(t'_j)$$

where

$$u_d = \sum_{j=1}^{M} t'_j\,[P_d(t'_j)]^2 \Big/ \sum_{j=1}^{M} [P_d(t'_j)]^2$$

$$V_d = \sum_{j=1}^{M} [P_d(t'_j)]^2 \Big/ \sum_{j=1}^{M} [P_{d-1}(t'_j)]^2 \qquad \underline{\text{except } V_1 = 0.}$$

$$[P_{ij}] = \begin{bmatrix} 1.0 & 1.0 & \cdots & 1.0 \\ P_{2,1} & P_{2,2} & \cdots & P_{2,\,d+1} \\ P_{3,1} & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ P_{M,1} & P_{M,2} & & P_{M,d+1} \end{bmatrix}$$

the desired polynomial is

$$y(t'_j) = \sum_{i=1}^{d+1} S_i \; P_i(t'_j)$$

the S_i are calculated as follows:

$$S_i = \sum_{j=1}^{M} X_j \; P_i(t'_j) \; / \; \sum_{j=1}^{M} [P_i(t'_j)]^2 \qquad i = 1, 2, \ldots d+1$$

or in matrix notation:

$$[S'_1, S'_2, \ldots S'_{d+1}] = [X_1, X_2, \ldots X_m] \begin{bmatrix} 1.0 & 1.0 & 1.0 \\ P_{2,1} & P_{2,2} \cdots & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ P_{M,1} & \cdots & P_{M,M} \end{bmatrix}$$

and

$$S_i = S'_i \; / \; \sum_{j=1}^{M} [P_{ij}]^2$$

To obtain the coefficients in the usual form:

$$P_i(t'_j) = \sum_{i=1}^{d+1} b_{i,\,d+1} \; (t'_j)^i$$

$$b_{i,k} = b_{i-1,k-1} - u_k b_{i,k-1} - V_k b_{i,k-2}$$

where

$$b_{i,k} = 1.0 \qquad i = k$$

$$= 0.0 \qquad i < 0 \text{ or } i > d+1$$

Yielding a lower triangular array with 1's down the diagonal

$$\begin{array}{lllll}
b_{1,1} & & & & \\
b_{1,2} & b_{2,2} & & & \\
b_{1,3} & b_{2,3} & b_{3,3} & & \\
b_{1,4} & b_{2,4} & b_{3,4} & b_{4,4} & \\
\vdots & & & & \\
b_{1,d+1} & & & & b_{d+1,\,d+1}
\end{array}$$

$$[S_1 \ldots S_{d+1}] \begin{bmatrix} 1.0 & & & \\ b_{12} & 1.0 & & 0 \\ b_{13} & b_{23} & 1.0 & \\ \vdots & & & \\ b_{1,d+1} & \cdots & b_{d,d+1} & 1.0 \end{bmatrix} = a_o, a_1, a_2, \ldots a_d$$

SUBROUTINE IDENTIFICATION

A. Title

OBSIN

B. Segment

PREPOD

C. Called by subroutine

LODOBS

FUNCTION

To subtract station biases, convert to internal units, and compute the time of the observations in minutes from midnight, day of epoch.

USAGE

The usage is essentially as described in the NRTPOD document, Reference 1, Section 5.3, the only difference being that the logic to handle range-rate observations has been eliminated.

SUBROUTINES USED

A. Library

—

B. Program

TIME Computes Julian date and minutes from midnight of epoch day

SUBROUTINE IDENTIFICATION

A. Title

OBSSRT

B. Segment

PREPOD

C. Called by subroutine

LODOBS

FUNCTION

To sort the observations by time with respect to the earliest time.

USAGE

The usage of this version of OBSSRT is essentially as described in the NRTPOD version, Reference 1, Section 5.3, with the exception that the logic which sorts the observations with respect to epoch has been eliminated.

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

PNCHVC

B. Segment

PREPOD

C. Called by subroutine

PREPOD

FUNCTION

To punch and print the state vector, STVEC, (ADBARV or X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$) on cards, the epoch TIME card and the TYPE card for use as input to NRTPOD in the NAMELIST format.

USAGE

A. Calling sequence

Call PNCHVC

B. Input

1. COMMON

/BLK1/CONS(30)	
KOUT	System output tape number
IOUT	System punch tape number
ITYPE	Type of output 1 = ADBARV 2 = X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$
/BLK2/WSTR(100)	
TIME(5)	Year, Month, Day, Hour, Minutes of epoch
SECS	Seconds of epoch
STVEC(6)	Polar or cartesian coordinates

2. Calling sequence

C. Output

1. COMMON

---

2. Calling sequence

---

3. Printed output

Print and punch STVEC, TIME and TYPE cards

D. Error/action messages

—

SUBROUTINES USED

A. Library

Input/Output

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title
PRNTC

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION

To print the block of program constants, station coordinates, and biases and to convert the station coordinates and biases to internal units.

USAGE

A. Calling sequence
Call PRNTC

B. Input

1. COMMON

/BLK1/CONS(30)		
KOUT	System output tape number	
CDEG	Degrees to radians	
CKMER	Kilometers to earth radii	
/BLK2/WSTR(100)		
STID		
STLAT	External units	degrees
STLONG	External units	degrees
STALT	External units	meters
RBIAS	External units	kilometers
ABIAS	External units	degrees
EBIAS	External units	degrees
TBIAS	External units	minutes

2. Calling sequence

C. Output

1. COMMON

/BLK2/WSTR(100)

STLAT	Internal units	radians
STLONG	Internal units	radians
STALT	Internal units	earth radii
RBIAS	Internal units	earth radii
ABIAS	Internal units	radians
EBIAS	Internal units	radians
TBIAS	Internal units	minutes

2. Calling sequence

—

3. Printed output

Program Constants/BLK1/CONS

Sensor coordinates and biases

D. Error/action messages

SUBROUTINES USED

A. Library

Input/output

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SCLOUT

B. Segment

PREPOD

C. Called by subroutine

PREPOD

FUNCTION

To scale the state vector (X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$ or ADBARV) from internal units (E.R., E.R./min, and radians) to external units (Km, Km/sec, and degrees).

USAGE

A. Calling sequence

Call SCLOUT

B. Input

1. COMMON

/BLK1/CONS(30)	
CKMER	Kilometers/earth radii
CDEG	Degrees/radian
ITYPE	Type of output desired 1 = ADBARV 2 = Cartesian
/BLK2/WSTR(100)	
STVEC(6)	State vector in internal units

2. Calling sequence

—

C. Output

1. COMMON

/BLK2/WSTR(100)	
STVEC(6)	State vector in external units

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title
SETCON

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION

To assign nominal values of program constants and clear memory. Nominal outputs are ADBARV state vector, epoch time, and Type = 1. Program is only entered once and is not entered for stacked cases.

USAGE

A. Calling sequence
Call SETCON

B. Input

1. COMMON
—

2. Calling sequence
—

C. Output

1. COMMON
/BLK1/CONS(30)

CKMER = 6378.165	Kilometers/earth radii
CDEG = 57.29577951	Degrees/radian
CWE = 4.37526906E - 3	Rotation rate
CELLIP = 1.0/298.3	Earth flattening
CDAYMN	Number of days in the month
(1) = 31.	January
(2) = 28.	February
(3) = 31.	March
(4) = 30.	April

(5)	= 31.	May
(6)	= 30.	June
(7)	= 31.	July
(8)	= 31.	August
(9)	= 30.	September
(10)	= 31.	October
(11)	= 30.	November
(12)	= 31.	December
CPI	= 3.1415926536	π
C2PI	= 6.2831853072	2π
KOUT	= 3	System output tape
IOUT	= 12	System punch tape
KIN	= 2	System input tape
ITYPE	= 1	ADBARV output
KDEG	= 4	4th degree polynomial

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

TPTOIN

B. Segment

PREPOD

C. Called by subroutine

PREPOD

FUNCTION

To convert the state vector from topocentric rectangular coordinates to geocentric inertial coordinates.

USAGE

A. Calling sequence

Call TPTOIN(STLAT, STLONG, STALT, COOR, STVEC)

B. Input

1. COMMON

—

2. Calling sequence

STLAT	Geodetic latitude of the observer
STLONG	Right Ascension of the observer
STALT	Altitude of the observer
COOR(6)	$(R, A, E, \dot{R}, \dot{A}, \dot{E})$ of the object from the observer

C. Output

1. COMMON

—

2. Calling sequence

STVEC(6)	Geocentric Inertial coordinates of the object $(X, Y, Z, \dot{X}, \dot{Y}, \dot{Z})$

D. Error/action messages

—

SUBROUTINES USED

A. Library
   SIN
   COS
   SQRT

B. Program
   —

EQUATIONS

Given R, A, E, $\dot{R}$, $\dot{A}$, $\dot{E}$ of an object observed from a station with coordinates:

φ = geodetic latitude

λ = right ascension

h = altitude

$$X = - R \cos E \cos A$$

$$Y = R \cos E \sin A$$

$$Z = R \sin E$$

$$\dot{X} = - \dot{R} \cos E \cos A + R\dot{E} \sin E \cos A + R\dot{A} \cos E \sin A$$

$$\dot{Y} = \dot{R} \cos E \sin A - R\dot{E} \sin E \sin A + R\dot{A} \cos E \cos A$$

$$\dot{Z} = \dot{R} \sin E + R\dot{E} \cos E$$

$$g_1 = \frac{a_e}{\sqrt{1 - (2f - f^2) \sin^2 \phi}} + h$$

$$g_2 = \frac{a_e (1-f)^2}{\sqrt{1 - (2f - f^2) \sin^2 \phi}} + h$$

$$X_1 = - g_1 \cos \varphi \cos \lambda$$

$$Y_1 = - g_1 \cos \varphi \sin \lambda$$

$$Z_1 = - g_2 \sin \varphi$$

$$\dot{X}_1 = - \omega_e Y_1$$

$$\dot{Y}_1 = + \omega_e X_1$$

$$\dot{Z}_1 = 0.0$$

$$\rho_X = X \sin\varphi \cos\lambda - Y \sin\lambda + Z \cos\varphi \cos\lambda$$

$$\rho_Y = X \sin\varphi \sin\lambda + Y \cos\lambda + Z \sin\lambda \cos\varphi$$

$$\rho_Z = - X \cos\varphi + Z \sin\varphi$$

$$\rho_{\dot{X}} = \dot{X} \sin\varphi \cos\lambda - \dot{Y} \sin\lambda + \dot{Z} \cos\varphi \cos\lambda$$

$$\rho_{\dot{Y}} = \dot{X} \sin\varphi \sin\lambda + \dot{Y} \cos\lambda + \dot{Z} \sin\lambda \cos\varphi$$

$$\rho_{\dot{Z}} = - \dot{X} \cos\varphi + \dot{Z} \sin\varphi$$

$$\dot{\rho}_X = \rho_{\dot{X}} - \omega_e Y$$

$$\dot{\rho}_Y = \rho_{\dot{Y}} + \omega_e X$$

$$\dot{\rho}_Z = \rho_{\dot{Z}}$$

$$X_I = \rho_X - X_1$$

$$Y_I = \rho_Y - Y_1$$

$$Z_I = \rho_Z - Z_1$$

$$\dot{X}_I = \dot{\rho}_X - \dot{X}_1$$

$$\dot{Y}_I = \dot{\rho}_Y - \dot{Y}_1$$

$$\dot{Z}_I = \dot{\rho}_Z - \dot{Z}_1$$

REFERENCES

1. "Functional Specifications of Lincoln Laboratory Orbit Determination Programs", 5452-6008-R0000, TRW Systems, 10 May 1966.

2. "ESPOD Mathematical and Subroutine Description", 8497-6065-RU-000, TRW Systems, 24 June 1964.

3. J. D. Brooks, "Variational System Associated with a Piecewise Linear Drag Parameter", 3122-49, TRW Systems, 2 May 1966.

4. P. A. Lavoie, "Specification of NRTPOD Modification— Functional Standard Deviation", 3412.9-19, TRW Systems, 12 April 1966.

5. P. A. Lavoie, "Covariance Matrix Output for NRTPOD" (6242-11) 3412.9-26, TRW Systems, 18 April 1966.

6. P. A. Lavoie, "Mathematical Formulation for the Inclusion of $\ddot{R}$ Data Into NRTPOD", 3412.9-18, TRW Systems, 11 April 1966.

7. P. A. Lavoie, "Radar Steering Ephemeris for NRTPOD", 3412.9-20, TRW Systems, 13 April 1966.

8. P. A. Lavoie, "Linear Constraints Formulation for NRTPOD", 3412.9-21, TRW Systems, 14 April 1966.

9. J. J. Goldman, "Preliminary Determination of Orbit Elements", 3412.9-11, TRW Systems, 29 March 1966.

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* TRW Systems under P.O. CC-939 to Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP None
3. REPORT TITLE Functional Specifications of NRTPOD Program Modifications	
4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)* Subcontract Report	
5. AUTHOR(S) *(Last name, first name, initial)* None given	

6. REPORT DATE February 1967	7a. TOTAL NO. OF PAGES 324	7b. NO. OF REFS 9
8a. CONTRACT OR GRANT NO. AF 19(628)-5167 b. PROJECT NO. Order 498 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 06242-6001-R001 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TR-67-64	

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency, Department of Defense

13. ABSTRACT

This document describes the six modifications which have been added to the NRTPOD program in the form of extended capabilities. The report is intended as an analyst's guide to the NRTPOD modifications as well as an operational handbook with input-output instructions. In addition, a separate stand-alone program is described which derives a preliminary estimate of an orbit and is designed to be used in conjunction with NRTPOD.

14. KEY WORDS

orbit determination programs
ESPOD
computer applications